

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1373

Analiza i usporedba sigurnosnih mehanizama u Internetu stvari

Filip Ptiček

Zagreb, lipanj 2021.

**SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Zagreb, 12. ožujka 2021.

DIPLOMSKI ZADATAK br. 1373

Pristupnik: **Filip Ptiček (0036491837)**

Studij: Informacijska i komunikacijska tehnologija

Profil: Telekomunikacije i informatika

Mentor: izv. prof. dr. sc. Marin Vuković

Zadatak: **Analiza i usporedba sigurnosnih mehanizama u Internetu stvari**

Opis zadatka:

Posljednjih godina razvoj bežičnih pristupnih mreža te energetski efikasnih ugradbenih računala ima za posljedicu pojavu malih uređaja koji nude mogućnosti kontrole, mjerena i praćenja okoline. Jedan od izazova Interneta stvari je sigurnost i privatnost korisnika te njihovih podataka. Taj izazov je posebno naglašen zbog heterogenosti uređaja i protokola za komunikaciju potrebnih kod implementacije sustava. Vaš je zadatak istražiti i usporediti postojeće tehnologije i protokole za komunikaciju te zaštitu podataka u okolini Interneta stvari. Na temelju istraživanja potrebno je osmisliti i implementirati rješenje na kojem ćete analizirati protokole u smislu osiguravanja osnovnih sigurnosnih zahtjeva.

Rok za predaju rada: 28. lipnja 2021.

Zahvaljujem se svome mentoru izv. prof. dr. sc. Marinu Vukoviću što mi je pružio priliku za izradu diplomskog rada na ovu temu, roditeljima Karmeli i Zlatku koji su mi omogućili obrazovanje na ovom fakultetu, KSET-u i ksetovcima s kojima sam proveo većinu svojih studentskih dana i Dunji na potpori tijekom izrade ovog rada.

SADRŽAJ

1. Uvod	1
2. Internet stvari	2
2.1. Definicija	2
2.2. Referentni model	3
2.2.1. Fizički uređaji i kontrolери	4
2.2.2. Povezanost	4
2.2.3. Računarstvo na rubu mreže	4
2.2.4. Akumulacija podataka	5
2.2.5. Apstrakcija podataka	5
2.2.6. Aplikacije	6
2.2.7. Suradnja i procesi	6
2.3. Izazovi	6
2.3.1. Heterogenost	7
2.3.2. Raspodijeljenost	7
2.3.3. Sigurnost	8
2.3.4. Privatnost	8
2.3.5. Integracija	8
2.4. Područja primjene	9
2.5. Trendovi	10
3. Sigurnost u Internetu stvari	12
3.1. Zahtjevi vezani uz sigurnost i privatnost	12
3.2. Sigurnosni propusti	13
3.2.1. Slabe, pogodljive ili tvrdo kodirane lozinke	13
3.2.2. Nesigurne mrežne usluge	14
3.2.3. Nesigurna sučelja ekosustava	15
3.2.4. Nedostatak mehanizama za sigurnosna ažuriranja	16

3.2.5. Upotreba nesigurnih ili zastarjelih komponenti	16
3.2.6. Nedovoljna zaštita privatnosti	17
3.2.7. Nesigurni prijenos i pohrana podataka	19
3.2.8. Nedostatak mogućnosti upravljanja uređajima	19
3.2.9. Nesigurne zadane postavke	20
3.2.10. Nedostatak fizičke sigurnosti	21
4. Analiza i usporedba protokola	23
4.1. Protokolni složaj Interneta stvari	23
4.2. Sloj uređaja	24
4.2.1. Senzorske pločice	24
4.2.2. Komunikacijski moduli	25
4.2.3. Analiza pristupnih uređaja	26
4.2.3.1. Raspberry Pi 4 Model B	26
4.2.3.2. Raspberry Pi Pico	27
4.2.3.3. Arduino Uno Rev3	28
4.2.3.4. Libelium WaspMote	28
4.2.3.5. ESP32	29
4.2.3.6. Pycom FiPy	30
4.2.4. Usporedba sigurnosnih mehanizama i primjena	30
4.3. Fizički sloj i sloj podatkovne poveznice	31
4.3.1. Analiza protokola	32
4.3.1.1. IEEE 802.11 Wi-Fi	32
4.3.1.2. Bluetooth Low Energy	35
4.3.1.3. LoRaWAN	38
4.3.2. Usporedba sigurnosnih mehanizama i primjena	43
4.4. Mrežni sloj	44
4.4.1. Analiza protokola	44
4.4.1.1. IP	44
4.4.2. Usporedba sigurnosnih mehanizama i primjena	44
4.5. Transportni sloj	45
4.5.1. Analiza protokola	45
4.5.1.1. TCP	45
4.5.1.2. UDP	46
4.5.2. Usporedba sigurnosnih mehanizama i primjena	46
4.6. Aplikacijski sloj	46

4.6.1.	Analiza protokola	46
4.6.1.1.	HTTP	46
4.6.1.2.	CoAP	49
4.6.1.3.	MQTT	50
4.6.2.	Usporedba sigurnosnih mehanizama i primjena	52
5.	Sustav za praćenje tjelesne temperature	53
5.1.	Korišteni razvojni alati i uređaji	53
5.1.1.	Golang	53
5.1.2.	Fiber	53
5.1.3.	React	54
5.1.4.	PostgreSQL	54
5.1.5.	Docker	54
5.1.6.	Nginx	54
5.1.7.	ESP32	54
5.2.	Arhitektura i dizajn sustava	55
5.2.1.	Baza podataka	56
5.2.2.	Back-end aplikacija	56
5.2.3.	Front-end aplikacija	57
5.2.4.	Uređaj za mjerjenje temperature	58
5.3.	Opis rada sustava	59
5.3.1.	Registracija korisnika	59
5.3.2.	Prijava korisnika	59
5.3.3.	Kreiranje korisnika	60
5.3.4.	Uređivanje korisnika	60
5.3.5.	Konfiguracija uređaja	60
5.3.6.	Očitavanje tjelesne temperature	60
5.3.7.	Pregled trenda kretanja temperature	60
5.4.	Sigurnosna analiza sustava	61
5.4.1.	Tajnost	61
5.4.2.	Cjelovitost	61
5.4.3.	Raspoloživost	61
5.4.4.	Vjerodostojnjost	61
5.4.5.	Neporicanje	62
5.4.6.	Odgovornost	62
5.4.7.	Pouzdanost	62

6. Zaključak	63
Literatura	64
Popis slika	68
Popis tablica	69

1. Uvod

Internet stvari sve više obuhvaća našu okolinu i sadašnjicu. Pojavom brzih bežičnih pristupnih mreža te sve manjih i učinkovitijih uređaja povezanost se usadila u sve aspekte naših života. Pametni domovi, pametni gradovi i industrija 4.0 samo su neki od primjera gdje Internet stvari nalazi svoju primjenu. Povezanošću uređaja i krajnjih korisnika olakšavaju se razni dnevni procesi te se pokazuje prilika za pojavom novih koji prije nisu bili mogući. Iako se pojam Interneta stvari pojavio u zadnjih dvadeset godina, potreba za umrežavanjem uređaja datira od samog početka Interneta.

Tako se prvi pametni uređaj povezan na mrežu našao na Sveučilištu Carnegie Mellon 1982. godine. Zbog iritiranosti prouzročene praznim automatom za pića, jedan je student odlučio spojiti automat na ARPANET, javna mreža preteča Internetu, kako bi mogao doznati trenutno stanje automata[2]. Iako taj postupak tada nije bio smatran revolucionarnom idejom, broj umreženih aparata danas ukazuje na suprotno.

Trend područja Interneta stvari je u brzom rastu te poprima sve veću primjenu u svim područjima socijalnog života i poslovnih procesa te je važno definirati i rastumačiti sigurnosne mehanizme vezane uz Internet stvari kako bi implementacijska rješenja krajnjim korisnicima pružala adekvatnu razinu sigurnosti i privatnosti.

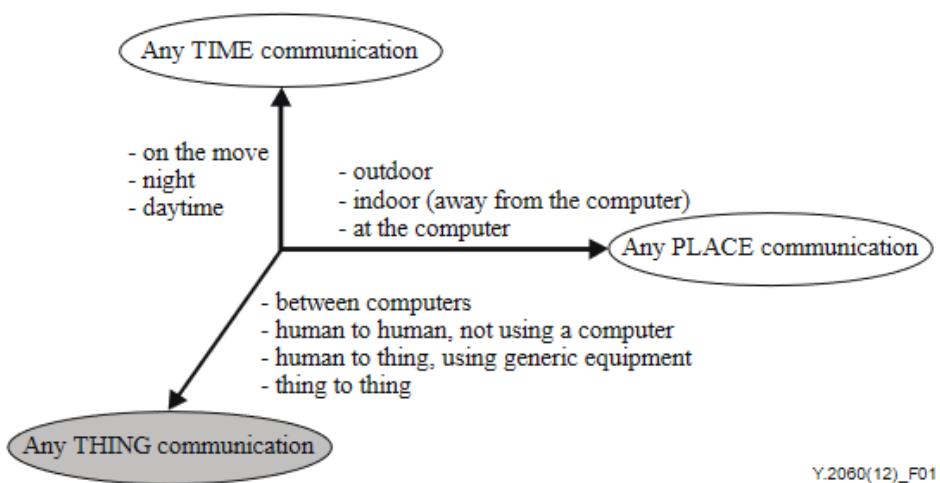
Ovaj rad primarno obrađuje sigurnosne mehanizme u Internetu stvari, uz glavni naglasak na informacijskoj sigurnosti, analizi i usporedbi sigurnosnih mehanizama protokola te implementaciji i sigurnosnoj analizi sustava za praćenje tjelesne temperature.

U sljedećem poglavlju će se definirati pojam, referentni model, izazovi, područja primjene i trendovi u Internetu stvari. Treće poglavlje obrađuje zahtjeve vezane uz sigurnost i privatnost te sigurnosne propuste u Internetu stvari. U četvrtom poglavlju se kroz protokolni složaj Interneta stvari analiziraju i uspoređuju protokoli svakog sloja uz naglasak na sigurnosnim mehanizmima protokola. Peto poglavlje opisuje implementaciju sustava za praćenje tjelesne temperature, arhitekturu sustava, korištene razvojne alate i uređaje te opisuje rad sustava i analizu sigurnosti sustava. Konačno u zaključku se utvrđuju implementacijski i sigurnosni izazovi oko razvoja sustava Interneta stvari.

2. Internet stvari

2.1. Definicija

The International Telecommunication Union (ITU-T) je specijalizirana agencija Ujedinjenih Naroda za informacijske i komunikacijske tehnologije. ITU-T definira Internet stvari kao globalnu infrastrukturu za informacijsko društvo, koja omogućava napredne usluge međusobnim povezivanjem (fizičkih i virtualnih) stvari na temelju postojećih i razvijajućih interoperabilnih informacijskih i komunikacijskih tehnologija. Iskorištavanjem identifikacije, prikupljanja podataka, obrade i komunikacijskih sposobnosti, Internet stvari u potpunosti upotrebljava mogućnosti povezanih stvari kako bi ponudio usluge za mnogo različitih primjena, uz osiguravanje sigurnosti i privatnosti. Sa šire perspektive, Internet stvari može biti percipiran kao vizija s tehnološkim i društvenim implikacijama[4]. Informacijske i komunikacijske tehnologije (ICT) pružaju komunikaciju u bilo koje vrijeme i na bilo kojem mjestu dok Internet stvari dodaje još jednu dimenziju gdje se radi o bilo kojoj stvari u komunikaciji. Te tri dimenzije komunikacije su prikazane na sljedećoj slici.

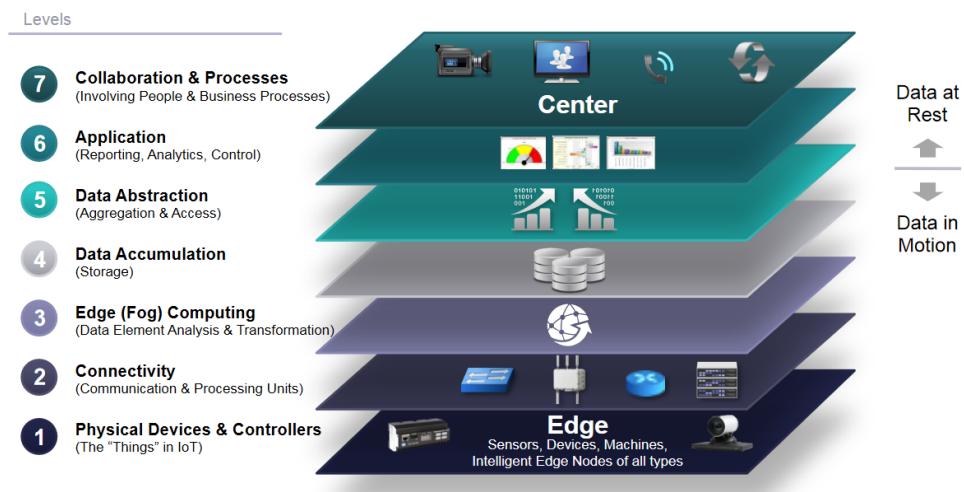


Slika 2.1: Nova dimenzija komunikacije predstavljena u Internetu stvari[4]

ITU-T također definira pojmove uređaja i stvari u kontekstu Interneta stvari. Uređaj je dio opreme s obaveznom mogućnošću komunikacije i neobaveznim mogućnostima opažanja, aktuacije te prikupljanja, pohrane i obrade podataka. Pojam stvari je definiran kao objekt u fizičkom svijetu (fizička stvar) ili u informacijskom svijetu (virtualna stvar) koji ima sposobnost da bude identificiran i integriran u komunikacijsku mrežu. Fizičke stvari postoje u fizičkom svijetu i imaju sposobnosti biti opažene, aktuirane i povezane, a virtualne stvari postoje u informacijskom svijetu i imaju sposobnosti biti spremljene, obrađene i pristupljene. Neki od primjera fizičkih stvari su: okolina, industrijski roboti, proizvodi i električna oprema, dok su primjeri virtualnih stvari multimediji sadržaji i programska podrška.

2.2. Referentni model

Ako promatramo Internet stvari kao jedan zaseban ekosustav, potrebno je definirati referentni model prema kojem možemo opisati sve dijelove sustava i njegove zahtjeve. S obzirom na to da je Internet stvari pojam koji opisuje povezanost stvari, a ne i konkretni referentni model, od pojave samog pojma predlagani su različiti modeli koji bi predstavljali cijeli spektar mogućnosti i zahtjeva Interneta stvari.



Slika 2.2: Ciscov referentni model Interneta stvari[5]

Ciscov referentni model Interneta stvari[5] definira višeslojni model od kojih svaki sloj definira terminologiju koja može biti standardizirana kako bi se stvorio globalni referentni okvir. Ovaj model ne definira lokalnost komponenata već opisuje zadatke koje svaki sloj obavlja kako bi se održala jednostavnost, omogućila skalabilnost i osigurala potpora. Model također definira funkcije koje su potrebne kako bi sustav Interneta

stvari bio kompletan. Na slici 2.2 prikazan je Ciscov referentni model Interneta stvari i njegovi slojevi. Put podataka između slojeva je dvosmjeran, dok je put kontrolnih informacija s viših slojeva prema nižim slojevima. Kod promatranja je put informacija u obrnutom smjeru, od nižih slojeva prema višim.

2.2.1. Fizički uređaji i kontroleri

Referentni model počinje prvim slojem: fizički uređaji i kontroleri koji mogu upravljati s više uređaja. Ovo je sloj koji opisuje stvari u kontekstu Interneta stvari i uključuje različiti raspon uređaja koji šalju i primaju informacije. Uređaji su različitih veličina, izgleda i namjene te potječe od različitih proizvođača. Kako bi se pojednostavila kompatibilnost, referentni model općenito opisuje razinu obrade potrebne od uređaja. Neke od osnovnih sposobnosti uređaja uključuju: pretvorbu analognih u digitalne signale, generiranje podataka i mogućnost da se uređajem upravlja i šalju upiti.

2.2.2. Povezanost

Komunikacije i povezanost su sadržani u drugom sloju. Najvažnija mogućnost ovog sloja je sposobnost pouzdanog i pravovremenog prijenosa informacija. Time se definira prijenos između uređaja i mreže, između mreža te između mreže i računarstva na rubu mreže (engl. *Edge Computing*) na trećem sloju. Jedan od ciljeva referentnog modela je da se sva komunikacija odvija putem postojećih mreža. Kako neki uređaji ne podržavaju protokol IP, potrebno je u mrežu uvesti prilaze (engl. *gateway*) koji će služiti kao posrednik između uređaja i ostatka mreže. Na ovom se sloju pojavljuje velika heterogenost komunikacijskih i pristupnih protokola koji uvelike ovise o željenoj namjeni uređaja prvog sloja. Ovaj sloj je usko povezan sa TCP/IP složajnim modelom koji sadrži protokole fizičkog sloja, sloja podatkovne poveznice te mrežnog, transportnog i aplikacijskog sloja.

2.2.3. Računarstvo na rubu mreže

Funkcija trećeg sloja je vođena potrebom za pretvaranjem mrežnog podatkovnog prometa u informacije koje su prikladne za pohranu podataka i za obradu na višim slojevima. Treći sloj je zadužen za obradu podataka i njihovu transformaciju. Jedno od načela ovog referentnog sloja je da se obrada podataka odvija što je ranije moguće i što bliže rubu mreže kako bi se smanjila potreba za odvijanjem obrade velikog skupa podataka na udaljenom i centralnom mjestu. Obrada na trećem sloju obuhvaća razne

primjere poput: evaluacije, formatiranja, proširivanja, dekodiranja, redukcije i procjene značenja podataka.

2.2.4. Akumulacija podataka

Mrežni sustavi su izgrađeni za pouzdani prijenos podataka. Prije četvrtog sloja podaci su u stanju prijenosa. Takvi podaci proizlaze iz prvog i prolaze kroz drugi i treći sloj. Kako u nekim slučajevima ne postoji potreba za trenutnom obradom tih podataka, oni dolaze do četvrtog sloja gdje se podaci spremaju u memoriju. Na ovom sloju su podaci trajni i nepromjenjivi te spremni za posluživanje višim slojevima referentnog modela. Četvrti sloj određuje jesu li podaci važni za više slojeve te, ako jesu, potrebno je osigurati načine posluživanja tih podataka zahtjevima viših slojeva. Određuje je li potrebno da podaci budu trajni, tj. treba li podatke spremiti na trajnu ili ih je dovoljno spremiti u radnu memoriju za kratkoročnu upotrebu. Određuje i kakav tip pohrane podataka je potreban: datotečni sustav, distribuirani datotečni sustav ili neki oblik baze podataka. Također određuje na koji način je organizirano spremanje podataka te je li potrebno podatke spojiti, preračunati i agregirati s prethodno spremljениm podacima. Ukratko, zadaća četvrtog sloja je da podatke bazirane na događajima pretvoriti u podatke nad kojima se rade upiti za potrebe viših slojeva.

2.2.5. Apstrakcija podataka

Funkcije apstrakcije podataka petog sloja su fokusirane na prikazivanje podataka i njihovu pohranu na način koji dozvoljava razvoj jednostavnijih, brzih aplikacija. Kako u modelu Interneta stvari postoji više uređaja koji generiraju podatke, tako postoje različiti razlozi zašto podaci nisu prisutni na istom podatkovnom spremištu: previše podataka za spremanje na jedno mjesto, uređaji su geografski odvojeni, a obrada je optimizirana lokalno, postoji potreba za različitim načinima obrade podataka te se koriste različiti načini akumulacije podataka. Zbog tih razloga peti sloj je zadužen za različite vrste obrade poput ujednačavanja različitih formata podataka iz različitih izvora, osiguravajući dosljednu semantiku podataka kroz različite izvore, potvrda o potpunosti podataka šestom aplikacijskom sloju, zaštita podataka korištenjem autorizacijskih i autentifikacijskih mehanizama te normaliziranja i indeksiranja podataka za brz pristup od strane aplikacija.

2.2.6. Aplikacije

Na šestom sloju se nalazi aplikacijski sloj koji obavlja interpretaciju informacija. Ovaj referentni model ne definira strogo aplikaciju. Aplikacije se razlikuju na temelju različitih tržišta, prirode podataka i poslovnih potreba. Primjeri različitih potreba su aplikacije koje su usredotočene na promatranje i prikupljanje podataka; neke aplikacije se koriste za kontrolu uređaja dok neke kombiniraju podatke s uređaja i drugih izvora. Aplikacije predstavljaju različite modele upotrebe, razvojne obrasce, korištenju razvojnu programsku podršku te krajnje kompleksnosti upotrebe. Neki primjeri aplikacija su povezani sa specijaliziranim industrijskim rješenjima, neke mobilne aplikacije obavljaju jednostavne interakcije, neke izrađuju izvješća za poslovne procese, analitičke aplikacije obrađuju i interpretiraju podatke važne za poslovne odluke, a neke aplikacije upravljaju i kontroliraju ostatak sustava. Ako su prijašnji slojevi dizajnirani pravilno, to će utjecati na količinu posla koje sama aplikacija mora raditi dok će to zauzvrat olakšati procese na sedmom sloju.

2.2.7. Suradnja i procesi

Zadnji sedmi sloj referentnog modela uključuje ljude i poslovne procese. Ljudi koriste aplikacije i pridružene podatke za svoje specifične potrebe. Često, više ljudi koristi iste aplikacije za različite svrhe. Tako cilj cijelog sustava Interneta stvari nije sama aplikacija nego ispomoći u radu ljudi. Aplikacije pomažu ljudima kako bi mogli obavljati različite poslovne procese uz odgovarajuće podatke u pravo vrijeme. Poslovni procesi nerijetko uključuju rad i komunikaciju između više ljudi. Ljudi surađuju i komuniciraju međusobno kako bi potpora Interneta stvari bila korisna. Zato ti procesi zahtijevaju više koraka koji obuhvaća više aplikacija. Stoga zadnji sloj predstavlja višu razinu od jedne aplikacije.

2.3. Izazovi

Internet stvari kao pojam i skup tehnologija donosi i određene izazove. S obzirom na to da je Internet stvari široko područje koje ima različita područja primjene, tako su i izazovi s kojima se susreće kod razvoja, planiranja i održavanja sustava brojni. Korištenjem pravilnih oblikovnih obrazaca postižu se bolja svojstva sustava te se olakšava daljnje održavanje i korištenje. U nastavku su nabrojani i opisani neki od izazova koji se pojavljuju u sustavima Interneta stvari.

2.3.1. Heterogenost

Heterogenost se pojavljuje na svakom implementacijskom koraku sustava Interneta stvari. Heterogenost uređaja, komunikacijskih, pristupnih i transportnih protokola, programske podrške i samih potreba korisnika. Naravno, ovakva vrsta heterogenosti se javlja zbog različitih potreba i radnih procesa. Uređaji koji se koriste imaju različite zahtjeve u vidu veličine, potrebe za vanjskim napajanjem te dometa komunikacijskih kanala, radilo se o žičanoj ili bežičnoj komunikaciji. Protokoli koji vrše komunikaciju između uređaja i nekog oblika prilaza također ovise o tim potrebama, radilo se o potrebi za velikim transportnim brzinama ili o korištenju radiokomunikacije niske snage kako bi se očuvala energija uređaja. Također, pouzdanost i sigurnost komunikacije ovisi o korištenju različitih transportnih protokola te kriptografskih algoritama. Na sve navedeno utječu i sami radni procesi i potrebe korisnika. Ovisno o tome, sami sustavi su osmišljeni za potrebe korisnika.

Heterogenost koja se javila u prvim razdobljima razvoja Interneta svari je povezana s nepostojanjem standardizacije i interoperabilnosti uređaja i postojećih programskih platformi. Tako su se na primjeru pametnih domova pojavili različiti pametni uređaji od kojih je svaki zahtijevao vlastitu programsku platformu za upravljanje zbog nepostojanja interoperabilnosti s centralnim upravljačkim platformama. Kroz vrijeme su se pojavili zajednički napori proizvođača i različitih standardizacijskih tijela da se ovisno o području primjene sustava Interneta stvari postigne standardizacija komunikacijskih protokola i semantike informacija kako bi se postigla bolja interoperabilnost.

2.3.2. Raspodijeljenost

Kako u sustavima Interneta stvari sudjeluju različiti uređaji različitih prostornih lokacija tako se javlja i potreba za raspodijeljenosti sustava. Uređaji mogu biti mobilni, što uvelike utječe na način pristupa krajnjim programskim platformama s kojima uređaji komuniciraju. Od dovođenja uređaja u sustav, upravljanja uređajima, promatranja uređaja i samih aplikacija preko kojih se ti postupci provode, sve se odvija raspodijeljenim putem. U nekim sustavima broj sudionika dostiže velike brojke te je kod takvih sustava važno da se paralelno i konkurentno mogu odvijati aktivnosti. Otpornost na kvar je još jedan od zahtjeva koji prati raspodijeljene sustave kako bi se omogućio nesmetan rad sustava. Kod ispada i kvarova, posljedice koje sustav Interneta stvari može imati na vanjski svijet su velike. Posebice u primjerima gdje se nadziru neki kritični sustavi poput industrijskih postrojenja. Također u pametnim domovima gdje su različiti kućanski aparati pretvoreni u pametne uređaje, kvarom poslužitelja mogu postati neupotrebljivi

te je važno da postoji redundantnost servisa s kojima uređaji komuniciraju.

2.3.3. Sigurnost

Sigurnost informacija i samih sustava je važan aspekt Interneta stvari u kojem sve više uređaja oko nas postaje umreženo. Problemi koji se javljaju su povezani s pokušajem brzog razvoja rješenja zbog konkurentnosti na tržištu, stavljanjem funkcionalnosti sustava u prvočitni plan te dostupnosti prema korisnicima što stavlja sigurnost tih sustava u drugi plan. Kako mnogi uređaji i cjelokupni sustavi uvelike imaju kritične zadatke, poput medicinskih uređaja ili automobila, kompromitacija istih zbog sigurnosnih propusta može imati negativne posljedice. Također neki od primjera napada na sustave nisu kako bi se napravila šteta sustavu, već za iskorištavanje procesne snage sustava u botnetovima. Sigurnosni propusti se mogu pojaviti na svakom dijelu sustava: uređajima, komunikacijskim kanalima, poslužiteljima, bazama podataka i korisničkim sučeljima. Neki od najvažnijih sigurnosnih zahtjeva na koje treba obratiti pozornost tijekom razvoja sustava Interneta stvari su obrađeni u trećem poglavlju.

2.3.4. Privatnost

Privatnost je uz sigurnost izazov s kojim se susreću mnogi sustavi Interneta stvari. Pametni uređaji koji se nalaze u našoj blizini imaju mogućnosti pratiti i bilježiti podatke o nama i našoj okolini. Kako korisnici priključuju sve više i više uređaja, tako je količina prikupljenih podataka veća, čime i povreda privatnosti može imati negativne posljedice na korisnika. Povrede te privatnosti mogu dolaziti u različitim oblicima. Od osobnih informacija koje mogu detaljno identificirati korisnika čime nastaje mogućnost napada u obliku krađe identiteta, medicinskih podataka ili pristupnih lozinki različitim servisima do kompromitacije sigurnosnih kamera što dozvoljava napadačima izravno praćenje korisnika. Narušavanjem privatnosti podataka pravnih osoba može doći do narušavanja poslovnih tajni, autentifikacijskih i autorizacijskih podataka za razne servise i sigurnosne sustave. Kako pristupiti zaštiti privatnosti je usko povezano sa sigurnošću sustava te je obrađeno u trećem poglavlju.

2.3.5. Integracija

Nove tehnologije donose i nove integracijske probleme. U slučaju Interneta stvari, ovaj izazov je veći zbog heterogenosti svih dijelova koji čine sustave. U slučaju heterogenosti sloja podatkovne poveznice i količine različitih protokola koji se koriste, postoji

problem uvođenja novih prijamnika i prilaza za te uređaje. Radi li se o niskofrekventnim rješenjima poput NFC-a ili protokolima poput ZigBee, postoji potreba korištenja posebnih čitača ili prijamnika, dok se taj problem ne pojavljuje kod protokola Wi-Fi čija pristupna točka već postoji u većini domova. Nakon toga dolazimo do komunikacijske integracije između uređaja i poslužitelja, tj. programskih platformi. Na koji način će se odvijati prijenos podatka, organizacija podataka te semantičko značenje tih istih podataka. Početkom razvoja rješenja Interneta stvari, prvobitno su svi sustavi imali vlastite pristupne aplikacije. S pojavom želje za automatizacijom i međusobnom komunikacijom između uređaja različitih proizvođača, počele su se razvijati platforme koje omogućuju interoperabilnost među sustavima te mogućnost upravljanja i praćenja uređaja putem jedne pristupne aplikacije. Kroz razvoj tih platformi, proizvođači su počeli integrirati nove i postojeće uređaje da budu kompatibilni s njima.

2.4. Područja primjene

Internet stvari ima široka područja primjene. U današnjem svijetu Internet stvari postaje sve prisutniji u svakom aspektu života ljudi i taj trend će se nastaviti. Od naših domova do industrijskih postrojenja, Internet stvari omogućava pojednostavljenje naših života i poslovnih procesa. U nastavku su navedena neka od glavnih područja primjene Interneta stvari i rješenja koja se koriste u tim područjima:

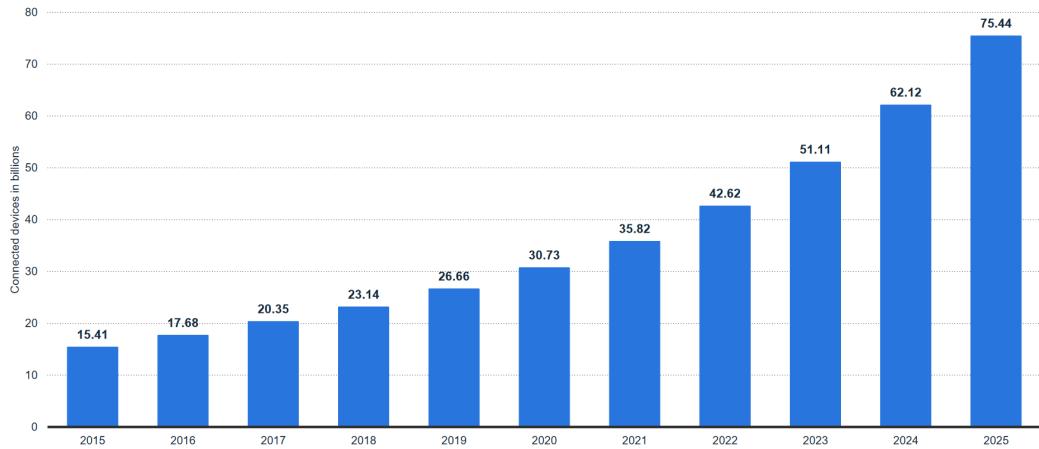
- Pametni dom
 - Pametna rasvjeta
 - Pametni kućanski aparati
 - Detekcija uljeza
 - Upravljanje energijom
- Pametni grad
 - Pametni parking
 - Upravljanje otpadom
 - Pametna rasvjeta
 - Reagiranje na hitne slučajeve
- Okoliš
 - Praćenje vremena
 - Praćenje zagađenja zraka

- Praćenje zagađenja bukom
- Detekcija požara
- Prodaja
 - Upravljanje inventarom
 - Pametni automati za prodaju
 - Pametne blagajne
 - Pametno plaćanje
- Logistika
 - Praćenje flote vozila
 - Praćenje pošiljaka
 - Dijagnostika vozila na daljinu
 - Generiranje i vremensko raspoređivanje voznih ruta
- Industrija
 - Dijagnostika strojeva
 - Praćenje dijelova proizvodnje
 - Automatizacija proizvodnih procesa
- Poljoprivreda
 - Pametno navodnjavanje
 - Praćenje usjeva
 - Automatizacija obrađivanja

2.5. Trendovi

Internet svari se pojavio kao zamisao za daljnju budućnost, ali je već danas sadašnjost u socijalnim i proizvođačkim aspektima svijeta. To može biti pripisano pristupačnjim, procesorski bržim, energetski učinkovitijim te sve manjim uređajima. Uz rast potražnje za integracijom takvih rješenja u različita područja, ona također kroz vrijeme postaju sve raznovrsnija. Trenutni broj umreženih uređaja Interneta stvari doseže brojku od 35.82 bilijuna uređaja. Pojavom novih tehnologija poput 5G mreža, koje dozvoljavaju još veće brzine i smanjeno vrijeme odaziva, taj trend će kroz godine još više rasti te su neke od projekcija za godinu 2025. u iznosu od 75.44 bilijuna uređaja[7]. Trend broja

umreženih uređaja Interneta stvari od 2015. godine i predikcije do 2025. godine su prikazane na sljedećoj slici.



Slika 2.3: Broj povezanih uređaja Interneta stvari[7]

Ako promatramo porast broja umreženih uređaja Interneta stvari, možemo zaključiti da uz to postoji i veliki tržišni potencijal. Korisnika je sve više te je procijenjeno da svaki posjeduje prosječno četiri uređaja Interneta stvari te da se u svijetu svake sekunde spoji 127 novih uređaja. Tako su i procjene ekonomskog utjecaja Interneta stvari na tržište između 3.9 i 11.1 bilijuna američkih dolara kroz različita područja primjene, uključujući tvornice, gradove, domove i prodaju[38]. Na sljedećoj slici su prikazane niske i visoke predikcije za ekonomski utjecaj Interneta stvari za 2025. godinu.



Slika 2.4: Potencijalni ekonomski utjecaj Interneta stvari kroz različita područja primjene za 2025. godinu. izražen u bilijunima američkih dolara[38]

3. Sigurnost u Internetu stvari

3.1. Zahtjevi vezani uz sigurnost i privatnost

Važnost sigurnosti i privatnosti je nezanemariva. Ta važnost se posebice očituje u sustavima Interneta stvari gdje je ponekad veći naglasak stavljen na funkcionalnost, što stavlja sigurnosne zahtjeve u drugi plan. Jednostavnii uređaji s malo procesorske snage imaju problema s obavljanjem zahtjevnijih kriptografskih algoritama. Studija iz 2019. godine je utvrdila da je od 56 milijuna ispitanih uređaja 91.5% razmjene podataka bilo nešifrirano[33]. Nedostatak iskustva u području razvoja Interneta stvari također utječe na slabu sigurnost uređaja jer se ponekad kod planiranja ne utvrde svi mogući sigurnosni propusti te zahtjevi koji su vezani uz sigurnost i privatnost. Ti zahtjevi su ključan dio očuvanja sigurnosti cjelokupnog sustava i privatnosti samih korisnika. Zahtjevi vezani uz sigurnost i privatnost su sljedeći[35]:

Tajnost: informacija nije dostupna ili je izložena neovlaštenim osobama, entitetima ili procesima.

Cjelovitost: točnost i potpunost informacije.

Raspoloživost: informacija je dostupna na zahtjev i omogućeno je njen korištenje od strane ovlaštenih osoba, entiteta ili procesa.

Vjerodostojnjost: osoba, entitet ili proces je zaista onaj kojim se predstavlja.

Odgovornost: obveza izvještavanja o aktivnostima i preuzimanja odgovornosti za njih.

Neporicanje: sposobnost dokazivanja događaja ili aktivnosti i osoba, entiteta ili procesa koji su ih pokrenuli ili u njima sudjelovali.

Pouzdanost: konzistentno ponašanje i rezultati.

3.2. Sigurnosni propusti

The Open Web Application Security Project® (OWASP) je neprofitna organizacija čiji je cilj napredak i poboljšanje računalne sigurnosti informacijskih sustava. OWASP svojim projektima otvorenog koda vođenim putem razvojne zajednice radi na poboljšanju sigurnosti Interneta.

OWASP Internet of Things Project je projekt osmišljen kako bi pomogao proizvođačima, programerima i potrošačima i omogućio im bolji uvid i razumijevanje sigurnosnih problema vezanih uz Internet stvari. Tako korisnici u bilo kojem dijelu razvojnog procesa mogu donositi bolje odluke kod razvoja, postavljanja i pristupanja tehnologijama Interneta stvari[10]. 2018. godine izlazi *OWASP IoT Top 10* lista koja reprezentira deset najčešćih ranjivosti sustava Interneta stvari. Svih deset sigurnosnih ranjivosti je navedeno u nastavku uz opis sigurnosnih praksi koje bi trebale sprječiti te ranjivosti i sigurnosne propuste.

3.2.1. Slabe, pogodljive ili tvrdo kodirane lozinke

Prvi navedeni sigurnosni problemi kod sustava Interneta stvari su vezni uz lozinke. Da bi se uređaju moglo pristupiti i naknadno ga konfigurirati, uređaji dolaze s korisničkim računima koji služe korisnicima kako bi ih mogli upariti sa željenim sustavom ili kako bi proizvođač mogao upravljati uređajem u slučaju pomoći korisnicima ili kod ažuriranja uređaja. Za pristup tom korisničkom računu uređaja potrebna je lozinka koju krajnji korisnik kod prve upotrebe treba postaviti. Navike korisnika su većinom da iskoriste njima dobro poznatu lozinku koju koriste i za svoje druge korisničke račune. Ako napadač dobije pristup jednoj njihovoј lozinki ima i pristup ostalim računima. Na taj način se pristup korištenim uređajima koji imaju isto korisničko ime ili e-mail adresu i lozinku uvelike olakšava. Korisnici imaju i naviku koristiti slabe lozinke koje su vrlo česte i jako lako pamtljive. Tako su neke od najčešće korištenih lozinka jednostavni nizovi numeričkih znakova ili nizovi znakova na tipkovnici poput: 123456, 123456789, qwerty, ili sam engleski prijevod lozinke (engl. *password*)[37]. Napadi na lozinke se provode putem takozvanih *brute force* napada. Kako je procesna snaga današnjih računala dosegla vrlo visoke brzine računanja, tako se jednostavne i kratke lozinke mogu pogoditi u vrlo kratkom vremenu.

Ovakvi propusti ne zaobilaze ni proizvođače samih sustava i uređaja. Kod proizvodnje, proizvođači na uređaje postavljaju iste lozinke za sve uređaje kako bi kod testiranja ispravnosti lakše pristupili istima. Jedan od najboljih pokazatelja takvog

pristupa su usmjerivači/modemi telekom operatera za pristup Internetu koji imaju postavljenu istu zadanu lozinku i korisničko ime poput "admin" ili "user" koju krajnji korisnici uređaja nikada ne promjene. Problem se također pojavljuje i u tvrdo kodiranim (engl. *hard coded*) lozinkama. Proizvođači postave takve lozinke na uređaje kako bi se uređaji mogli nesmetano povezati s vanjskim servisima, kako bi se proizvođači povezali na uređaj zbog otklanjanja pogrešaka ili kao način za vanjsko upravljanje uređajima. Ako napadač ima fizički pristup uređaju on može skenirati memoriju i pomoći raznih alata pronaći lozinku spremljenu na samom uređaju. A kako proizvođači najvjerojatnije koriste istu lozinku za sve iste modele uređaja, napadač ima lak način za pristup i ostalim istim uređajima.

Kako bi se sprječila ova vrsta ranjivosti, neke od sigurnosnih mjer koje bi se trebale pratiti su sljedeće. Korisnici bi kod prve upotrebe uređaja trebali promijeniti zadalu lozinku koristeći duge, kompleksne i jedinstvene nizove znakova. Najjednostavniji način za postići te zahtjeve je korištenjem upravitelja lozinkama. Oni daju mogućnost generiranja lozinki uz mogućnost spremanja istih, bez potrebe da korisnik mora pamtitи sve jedinstvene i duge lozinke. Što se tiče zahtjeva od strane proizvođača, oni bi trebali razriješiti bolje načine upravljanja uređajima kako bi se izbjeglo korištenje istih ili čak tvrdo kodiranih lozinka za pristup uređaju ili vanjskim servisima. Također, proizvođači bi trebali upozoriti korisnika kod uspostave uređaja da promjeni zadalu lozinku.

3.2.2. Nesigurne mrežne usluge

Uređaji Interneta stvari koriste razne mrežne usluge kako bi mogli komunicirati s vanjskim servisima. Kako je moguće pristupiti tim uređajima putem Interneta, potrebno je pravilno osigurati sigurnost tih mrežnih usluga koje se izvršavaju. Neautoriziran pristup preko usluga iskorištavajući zadane lozinke, otvorene mrežne priključke te nepravilno podešeni vatrozidi dozvoljavaju napadaču da dobije pristup uređajima i poslužiteljima. Takvi napadi dozvoljavaju izvršavanje malicioznog koda, iskorištavanje uređaja za botnet, krađu podataka ili onesposobljavanje sustava.

Neke od sigurnosnih mjer koje se mogu poduzeti za osiguravanje mrežnih usluga su:

- korištenje zasebne lokalne mreže za sve pametne uređaje,
- spajati uređaje na isključivo sigurne mreže,
- instaliranje regularnih softverskih ažuriranja,

- isključivanje svih usluga koje pružaju vanjski pristup uređaju,
- isključivanje nepotrebnih mrežnih priključaka i usluga,
- isključivo korištenje protokola koji podržavaju šifriranje.

3.2.3. Nesigurna sučelja ekosustava

Nesigurna web sučelja, pozadinski API-jevi, servisi u oblaku i mobilna sučelja koja dozvoljavaju komunikaciju i interakciju s uređajem čine sveukupni ekosustav Interneta stvari. Kompromitacija bilo kojeg dijela sustava može uzrokovati i kompromitaciju cjelokupnog sustava. Ranjivost kod načina autorizacije i autentifikacije između uređaja i poslužitelja ili korisnika mobilnih i web aplikacija i poslužitelja su jedan od vektora napada na sustav. Također, nedostatak ili korištenje slabog šifriranja kod komunikacije može uzrokovati da napadač presretne i iskoristi sakupljene informacije za napad. Nedostatak pravilnog filtriranja ulazno/izlaznih podataka može dovesti do napada poput SQL injekcije. Još jedan projekt OWASP organizacije je *OWASP Top 10 Web Application Security Risks* koji nudi popis najčešćih ranjivosti za web i mobilne aplikacije. Nesigurna sučelja ekosustava imaju direktnu poveznicu s tim ranjivostima koje su:

- injekcije (SQL, NoSQL, OS, LDAP),
- neispravna autentifikacija,
- izlaganje osjetljivih podataka,
- XML External Entities(XXE) napadi,
- neispravna autorizacijska kontrola,
- pogrešna konfiguracija servisa,
- Cross-Site Scripting(XSS),
- nesigurna deserijalizacija podataka,
- korištenje biblioteka i komponenta s poznatim sigurnosnim ranjivostima,
- nedovoljno korištenje logova i praćenja sustava.[8]

Pravilno podešavanje autorizacije i autentifikacije korisnika, ali i uređaja, je najvažniji način osiguravanja raznih sučelja ekosustava. Filtriranje ulaznih i izlaznih podataka spriječava napade injekcijom, a pravilno podešavanje poslužitelja da koriste kriptografske algoritme za sve servise osigurava privatnu i sigurnu komunikaciju. Kroz cijeli ekosustav je potrebna i uspostava bilježenja i praćenja sustava kako bi se na vrijeme otkrila nepravilna ponašanja unutar samog sustava.

3.2.4. Nedostatak mehanizama za sigurnosna ažuriranja

Kroz vrijeme, za programska rješenja koja se trenutno koriste na uređaju će se pronaći ranjivosti. Kako bi se na vrijeme i jednostavnim putem mogli spriječiti napadi koji iskorištavaju te ranjivosti potrebna su nam softverska ažuriranja, kao i ažuriranja samog ugrađenog programa (engl. *firmware*) uređaja. Ako ne postoji način kojim dovodimo takva sigurnosna ažuriranja na uređaj, postoji rizik kompromitacije uređaja. Također ako su i implementirani načini sigurnosnih ažuriranja, potrebno je pridodati pažnju na način te implementacije ažuriranja. Ako se ne provjeravaju digitalni potpisi izvora ažuriranja, moguće je na uređaj poslati maliciozno ažuriranje koje će kompromitirati uređaj. Potrebno je i koristiti sigurne načine prijenosa tih ažuriranja poput šifriranja upotrebljavanog komunikacijskog kanala.

Trenutnim trendom brzog razvoja novih uređaja, proizvođači često ne daju dovoljno dugi period sigurnosnih ažuriranja. Tako će se dogoditi da proizvod nakon manje od dvije godine prestane dobivati ažuriranja te će pasti odluka na korisnika o tome hoće li kupiti novi uređaj ili riskirati kompromitaciju trenutnog. Najbolji pokazatelj toga su pametni telefoni od kojih većina tijekom svog perioda upotrebe dobije samo nekoliko sigurnosnih ažuriranja prije nego bude zapostavljena od strane proizvođača.

Kako bi se uređaji zaštili od budućih napada zbog novootkrivenih sigurnosnih propusta, potrebno je korisnicima uređaja pružati dugotrajna i česta sigurnosna ažuriranja. Prijenos ažuriranja je neophodno prenosi korištenjem sigurnih komunikacijskih kanala koji su šifrirani. Ažuriranjima koja su dostigla na uređaj je potrebno validirati izvor, provjeriti odgovara li digitalni potpis izvoru od kojeg bi trebalo stići ažuriranje te je potrebno i validirati samo ažuriranje kako bi se izbjeglo moguće umetanje malicioznog koda.

3.2.5. Upotreba nesigurnih ili zastarjelih komponenti

Nadovezano na nedostatak mehanizama za sigurnosno ažuriranje, peti po redu od sigurnosnih propusta je upotreba nesigurnih ili zastarjelih komponenti. Mnogi sustavi Interneta stvari kao dio svojeg programskog rješenja sadrže otvoreni kod koji održava zajednica koja nije direktno povezana s proizvođačem. Kada se otkrije ranjivost na nekom od korištenih otvorenih rješenja, proizvođač ili čeka na sigurnosnu zakrpu, ili će u najboljem slučaju sam riješiti sigurnosni propust te ga objaviti kako bi doprinio razvoju otvorenog rješenja. Nakon što sigurnosna zakrpa bude razvijena, potrebno je ažurirati sve uređaje ili dijelove sustava koji su ugroženi tim sigurnosnim propustom.

Ako govorimo o Internetu stvari u proizvođačkoj industriji, takozvanoj Industriji

4.0, upotreba zastarjele programske podrške koja je potrebna zbog jako specifičnih uređaja za proizvodnju, čija zadnja verzija zna datirati i više od deset godina, nije rijetka. Uvođenje takvih uređaja u sustave Interneta stvari također utječe na sigurnost i integritet cjelokupnog sustava, a ugrožavanje jednog uređaja može dovesti do napada na cijeli lanac opskrbe. Kod upotrebe gotovih proizvoda poput senzora, videokamera ili pametne rasvjete te integracijom istih u postojeći sustav, treba obratiti pozornost na dostupnost sigurnosnih ažuriranja te stanja uređaja za kojeg proizvođač još uvijek nudi sigurnosnu podršku.

Kod planiranja razvoja sustava Interneta stvari potrebno je uzeti u obzir trenutno, a i buduće stanje razvojne i sigurnosne podrške vanjske programske potpore i komponenti sustava. Najbolji način za sprječavanje sigurnosnih propusta je korištenje vlastito razvijene programske potpore ili korištenje dobro podržanih vanjskih biblioteka otvorenog koda s jakom i aktivnom razvojnom zajednicom. Upotrebu zastarjelih uređaja bez sigurnosne podrške proizvođača ili s potporom koja uskoro dostiže krajnji period (engl. *end of life*) je potrebno izbjegavati. Nakon puštanja sustava u produkciju, važno je nadziranje i praćenje vijesti vezanih uz sigurnosne propuste upotrebljenih komponenti i programske podrške kako bi se na vrijeme mogla spriječiti kompromitacija sustava. Sve ovo nije moguće ako bilo koji dio ustava nema implementirane mehanizme za sigurnosna ažuriranja. Ako neka od komponenti dostigne svoj krajnji period ažuriranja, potrebno je tu komponentu ukloniti i zamijeniti ju drugom čija su sigurnosna ažuriranja još uvijek podržana.

3.2.6. Nedovoljna zaštita privatnosti

Uloga Interneta stvari je djelom prikupljanje različitih podataka i mjerena. Neki od tih podataka su osobne prirode za korisnika, poput medicinskih podataka ili zvukovnih i video zapisa. Kompromitacija takvih privatnih podataka može negativno utjecati na sigurnost korisnika. Prostor na kojem se privatnost korisnika može narušiti je od samog uređaja koji prikuplja podatke, preko komunikacijskih kanala preko kojih se podaci šalju, do samih krajnjih servisa koji primaju i obrađuju te podatke, a zatim ih spremaju u baze podataka na poslužiteljima. Nedovoljna zaštita privatnosti je zapravo rezultat svih ostalih sigurnosnih ranjivosti nabrojanih u ovom odjeljku.

Za očuvanje privatnosti korisnika i načine obrade podataka korisnika u Europskoj uniji postoji uredba donesena od strane Europske unije pod nazivom *Opća uredba o zaštiti podataka(GDPR) (EU) 2016/679[6]*. Cilj uredbe je omogućiti građanima Europske unije veću kontrolu i uvid u podatke koji se prikupljaju. Tako građani mogu tražiti

brisanje svojih podataka i povećava se odgovornost pravnih osoba koje te podatke prikupljaju. Odgovornost se postiže mogućim sankcijama nametnutim ako se utvrdi povreda podataka građana. Prikupljanje podataka je moguće uz izrazitu privolu građana korisnika čime se zabranjuje bilo kakvo prikupljanje podataka bez pristanka.

Šifriranje komunikacijskih kanala nekada ne osigurava i privatnost korisnika. Kako bi pametni uređaji mogli komunicirati s krajnjim poslužiteljima, koji mogu mijenjati svoju odredišnu adresu, koriste se domenska imena. Za razlučivanje tih adresa u brojčane IP adrese koristi se protokol DNS(engl. *Domain Name System*). Kada uređaji rade DNS upite, u sadržaju upita se prikazuje i domena upita u nešifriranom formatu. Na taj način napadač može iz konteksta upita zaključiti koji uređaji proizvođača se nalaze u mreži korisnika. Za neke uređaje je moguće zaključiti i sam tip, a ne samo proizvođača. U sljedećoj tablici možemo vidjeti uređaje i DNS upite koje proizvode:

Tablica 3.1: Primjer DNS upita napravljenih od strane uređaja [31]

Uredaj	DNS upiti
Nest Security Camera	nexus.dropcam.com oculus519-vir.dropcam.com pool.ntp.org
Amazon Echo	ash2-accesspoint-a92.ap.spotify.com audio-ec.spotify.com device-metrics-us.amazon.com ntp.amazon.com pindorama.amazon.com softwareupdates.amazon.com

Još jedan način na koji se može zaključiti o trenutnoj aktivnosti korisnika u vlastitoj mreži je i broj paketa koji se šalje u danom trenutku van mreže kao i njihova periodičnost. Ako se radi o uređaju koji ima mogućnosti virtualnog asistenta, moguće je imati uvid u to kada je korisnik imao interakciju s uređajem. Također, kod uređaja koji prate spavanje korisnika, broj razmijenjenih paketa drastično se poveća kada korisnik spava[31].

Kako najbolje očuvati privatnost korisnika je pitanje s kojim još uvijek mnogi proizvođači imaju problema. To se očituje u ostalim navedenim sigurnosnim propustima u ovom odjeljku. Zakonskim regulativama postiže se veća svijest o bitnosti zaštite podataka te se samim time proizvođači tjeraju na bolje prakse očuvanja podataka. Neki od osnovnih načina zaštite korisničkih podataka su:

- šifriranje podataka u svakom aspektu sustava,
- prikupljanje samo nužnih podataka,
- anonimiziranje korisnika,
- bolja kontrola i uvid u podatke za korisnike.

3.2.7. Nesigurni prijenos i pohrana podataka

Podatke koji nisu šifrirani vrlo je lako moguće iščitati. Kriptografijom se postiže sigurnost i privatnost podataka. Kako bi se to postiglo, podatke je potrebno šifrirati u svakom koraku njihova nastajanja, prijenosa, obrade i spremanja. Korištenje samih kriptografskih algoritama ne rezultira uvijek i zaštitom podataka. Neki kriptografski algoritmi koriste ključeve nedovoljne duljine pa je potrebno malo vremena da se dešifriraju. Najveći sigurnosni propusti u nedavnoj povijesti povezani su direktno s neadekvatnim kriptografskim algoritmima ili općenitim nedostatkom šifriranja, čime su ugroženi osobni podaci i lozinke korisnika[34]. Pozornost se treba posvetiti i kontroli pristupa podacima kako neautorizirani korisnici ne bi mogli pristupiti nedozvoljenim podacima.

Osnovne sigurnosne mjere koje bi se trebale osigurati su:

- šifriranje podataka,
- pravilno korištenje PKI-a (engl. *public key infrastructure*),
- kontrola pristupa podacima,
- korištenje sigurnih protokola za prijenos podataka,
- provjera korištenih kriptografskih algoritama za ranjivosti,
- korištenje dugih kriptografskih ključeva.

3.2.8. Nedostatak mogućnosti upravljanja uređajima

Nemogućnost upravljanja uređajima ima posljedicu da uređaji u slučaju otkrivenih sigurnosnih propusta ne mogu biti ažurirani, da uređaje nije moguće na jednostavan način otkloniti i uvesti u ekosustav te naknadno proširivati njihove mogućnosti. Zato je jedan od najvažnijih sigurnosnih zadataka u ekosustavima Interneta stvari upravljanje uređajima kroz njihov životni ciklus. Ako neautorizirani uređaji budu uvedeni u ekosustav, imat će mogućnost dobivanja pristupa ostalim komponentama ekosustava te nadgledanja mreže i presretanja prometa i informacija.

Zbog heterogenosti trenutnih implementacijskih rješenja, jedinstven način upravljanja uređaja je također jedan od problema koji se pojavljuju. Ako imamo više uređaja od kojih svaki zahtjeva svoju platformu i drugačiji način upravljanja, stvara se problem da neki uređaji koji ne zahtijevaju konstantnu pozornost ostanu zaboravljeni, nenađizani i neažurirani.

Potrebno je imati implementirane načine upravljanja, nadzora i ažuriranja uređaja prisutnih u sustavu. Otkrivanje i identifikacija uređaja je bitan korak u nadgledanju i zaštiti cijelog ekosustava. Heterogenost implementacijskih rješenja uređaja je još uviјek problem s kojim se integratori rješenja susreću, ali kako cijelo područje sazrijeva dolazimo do različitih platformi koje nude integraciju njih svih u jedinstveni ekosustav. Stoga je kod planiranja sustava potrebno uzeti u obzir uređaje kojim je moguće jedinstveno upravljati kako bi se izbjeglo zanemarivanje uređaja.

3.2.9. Nesigurne zadane postavke

Zadane postavke na pametnim uređajima povezane su uz nekoliko primjera. Takav propust se može očitovati kod univerzalno zadanih lozinki, tvrdo kodiranih lozinki ili zadanih postavki programske podrške uređaja. Univerzalne zadane lozinke se pojavljuju kao najjednostavniji način prvobitnom pristupu uređaju umjesto nekog drugog načina uspostave uređaja. Takav pristup se mora spriječiti navođenjem upozorenja ili obaveznim postupkom promjene lozinke kod prvobitnog postavljanja uređaja. Tvrdo kodirane lozinke za pristup uređajima je problem koji kod fizičkog ili vanjskog pristupa uređaju može lako dovesti do kompromitacije cijelog sustava te je korištenje takvih lozinka i načina pristupa potrebno izbjegavati. Dužnost proizvođača je da sve nepotrebne i potencijalno nesigurne postavke programske podrške koje mogu dovesti do sigurnosnih propusta tijekom testiranja i sigurnosne revizije uoči i onemogući. To uključuje i sve metode koje su se koristile za testiranje i otklanjanje pogrešaka tijekom razvoja.

Ovaj sigurnosni propust nije isključivo vezan uz uređaje. Nesigurne zadane postavke se javljaju i na poslužiteljima te ostaloj opremi koja sudjeluje u cijelom lancu komunikacije. Usluge koje se javljaju kao zadane na operativnim sustavima poslužitelja ponekad su nepotrebne za rad sustava. Takve usluge mogu imati zadane postavke koje dozvoljavaju jednostavan ili nesiguran pristup poslužitelju. Ovakav tip ranjivosti se nadovezuje na propust nesigurnih mrežnih sučelja. Jedan od takvih primjera je konfiguracija vatrozida mreže koja po zadanim postavkama može dozvoliti nesmetan doljev vanjskog prometa prema lokalnoj mreži.

Sprječavanju sigurnosnih propusta vezanih uz nesigurne zadane postavke se treba

pristupiti iz dva smjera. Prvi je od strane korisnika. Ako uređaj dolazi s općenitom zadanim pristupnom lozinkom, korisnika se treba obavijestiti da promjeni lozinku. Drugi je sa strane proizvođača, koji treba ukloniti sve zadane postavke vezane uz testnu okolinu i lako pristupanje uređaju, poput tvrdo kodiranih lozinki, prije nego što se uređaj stavi u upotrebu. Kod uspostave poslužitelja potrebno je obratiti pozornost na konfiguraciju mrežnih usluga koje dozvoljavaju pristup i upravljanje samim poslužiteljem. To se odnosi i na sve mrežne uređaje koji se nalaze u lokalnoj mreži uređaja kako bi komunikacija bila sigurna i sprječavala neautorizirani vanjski pristup.

3.2.10. Nedostatak fizičke sigurnosti

Uređaji koji se koriste u Internetu stvari su u nekim slučajevima postavljeni na širokim, raspršenim i nenadziranim područjima poput polja ili šuma. Takvim uređajima je potrebna fizička sigurnost kako bi se spriječili napadi direktnog pristupa prvobitno uređaju, a zatim i napadi na ostatak sustava. Takvi uređaji postavljeni na otvorenom nalaze se u zaštitnim kućištima pa je prvi korak napada otvaranje tog kućišta. Zato je potrebno zaštiti kućišta te implementirati načine otkrivanja neovlaštenog pristupa (engl. *anti-tampering detection*). Ako napadač uspješno fizički pristupi uređaju, postoji nekoliko načina pristupa informacijama ili radu uređaja. Informacije se često na takvim uređajima spremaju na memorijske kartice iz kojih je izvlačenje spremljenih informacija moguće ukoliko sam sadržaj nije šifriran. Takvim načinom napada se mogu izvući lozinke ili privatni ključevi koje uređaj koristi za pristup vanjskim servisima. Na uređajima se također znaju nalaziti pristupni priključci poput USB ili serijskih priključaka. Ako ne postoji način autorizacije pristupnog korisnika, moguća je kompromitacija samog rada uređaja. Ti priključci se koriste za testiranja te se kod stavljanja uređaja u upotrebu često ne onesposobe.

Kod takvih fizičkih napada ponekad nije cilj kompromitacija sustava već samo onesposobljavanje uređaja za obavljanjem njihovog zadatka. Ako uređaj obavlja zadatke poput nadziranja prostora pomoću senzora za požar, dima ili pokreta, napad na takav uređaj može fizički našteti stvarima poput različitih industrijskih pogona i osiguranih prostora te nainjeti veliku financijsku štetu.

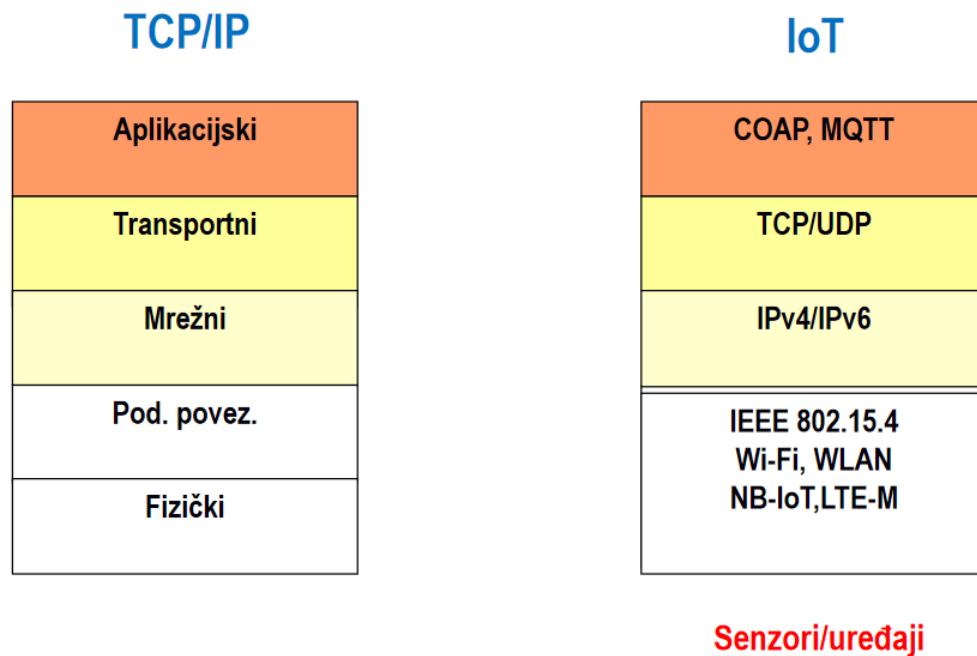
Kako bi se spriječio ili otežao fizički pristup uređajima, potrebno je koristiti zaštitna kućišta koja sprječavaju takve napade. Drugi sloj zaštite je korištenje mehanizama za otkrivanje neovlaštenog pristupa uz mogućnost obavještavanja korisnika o pristupu. Kod samih fizičkih uređaja potrebno je koristiti šifriranje memorije kako bi se spriječilo čitanje podataka s uređaja. Sljedeći korak zaštite je uklanjanje i onemogućavanje

svih nepotrebnih priključaka za pristup radu uređaja. Ako postoji potreban priključak za pristup, pristupanje je potrebno omogućiti samo autoriziranim korisnicima korišteњem kriptografskih ključeva ili lozinka.

4. Analiza i usporedba protokola

4.1. Protokolni složaj Interneta stvari

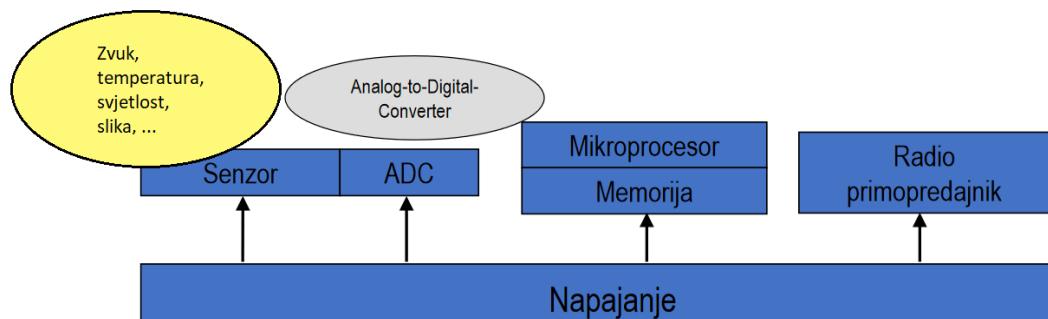
Protokolni složaj Interneta stvari se pobliže temelji na TCP/IP složaju koji podrazumijeva 5 slojeva u svom modelu. Ti slojevi su: fizički sloj, sloj podatkovne poveznice, mrežni, transportni i aplikacijski sloj. Svaki od tih slojeva sadrži različite protokole koji se koriste za komunikaciju na temelju fizičkih, komunikacijskih, identifikacijskih, pristupnih ili semantičkih karakteristika. Na sljedećoj slici su prikazani slojevi složaja uz primjere nekih protokola prisutnih u tom složaju. Uredaji i senzori su navedeni kao multi sloj jer su temelj svakog sustava Interneta stvari.



Slika 4.1: Protokolni složaj Interneta stvari[35]

4.2. Sloj uređaja

Sloj uređaja je bazni sloj u Internetu stvari. Umrežavanje tih uređaja i korištenje njihovih senzorskih i aktuatorских sposobnosti nam omogućava razvoj cijelokupnog sustava Interneta stvari. Uređaji se sastoje od nekoliko komponenti: napajanja, radio primopredajnika, senzora, analogno-digitalnih pretvornika koji dolaze kao zasebni moduli te mikroprocesora, memorije i razvojne pločice kao jedan jedinstveni uređaj. Postoje i već potpuno integrirani uređaji koji sadrže sve prije spomenute module na jednoj razvojnoj pločici. Prikaz modula uređaja nalazi se na sljedećoj slici.



Slika 4.2: Moduli uređaja Interneta stvari[35]

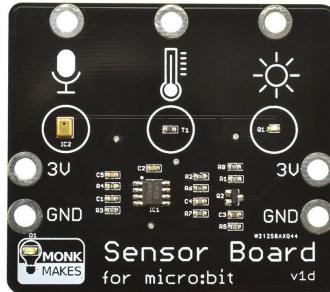
Kod upotrebe uređaja Interneta stvari postoji nekoliko zahtjeva koji se razlikuju s obzirom na primjenu. Neki od zahtjeva su: izrazito male dimenzije, mala potrošnja energije, niska cijena i umrežavanje na načelu samoorganizacije. Male dimenzije su potrebne kako bi uređaji djelovali neinvazivno na prostor u kojem se nalaze. Mala potrošnja energije je bitan zahtjev jer je dio uređaja mobilan te nema stalan izvor napajanja, već koristi baterije koje nemaju veliki kapacitet ili solarne ćelije čija stopa punjenja nije visoka. Zbog energetske učinkovitosti, uređaji imaju ograničenu procesorsku moć te je memorijski kapacitet ograničen. Najveća potrošnja energije se dogaža na komunikacijskim modulima čija snaga signala otpada s kvadratom udaljenosti. Svi ti zahtjevi se trebaju uzeti u obzir tijekom planiranja kako bi se mogao utvrditi potreban uređaj za namijenjenu primjenu.

U nastavku će biti opisane senzorske pločice, komunikacijski modul te analizirani pristupni uređaji od kojih su neki potpuno integrirani kao jedna razvojna pločica.

4.2.1. Senzorske pločice

Uloga senzorskih pločica je očitavanje vanjskih fizičkih stvari što je jedna od glavnih mogućnosti sustava Interneta stvari. Senzori prisutni na pločicama omogućuju opaža-

nje različitih vanjskih fizičkih pojava poput temperature, prisutnosti, pokreta, zvuka, koncentracije plinova, magnetskih polja i protoka vode. Integracija tih pločica s mikrokontrolerima omogućuje procesiranje tih opažanja. Primjer senzorske pločice s mogućnošću očitanja razine buke, temperature i luminacijskog intenziteta je prikazan na sljedećoj slici.



Slika 4.3: Senzorska pločica s mogućnošću očitanja razine buke, temperature i luminacijskog intenziteta[27]

4.2.2. Komunikacijski moduli

Komunikacijski moduli omogućuju uređajima da se umreže u ostatak sustava, čime se ostvaruje glavni zahtjev u Internetu stvari što je povezanost i mogućnost komunikacije. Moduli potrebni za komunikaciju ovise o fizičkom sloju i o sloju podatkovne poveznice. Fizički sloj uvjetuje fizički medij u kojem se podaci prenose. Žičani mediji su: bakrena žica u kojoj se podaci prenose elektronima i optički kablovi gdje se informacije prenose fotonima. Bežični mediji su: optički, koji ne koriste kablove već usmjerene laserske snopove koji putuju kroz zrak i radio valove čije različite frekven-cije koriste različiti protokoli sloja podatkovne poveznice.

Sloj podatkovne poveznice čine protokoli koji u osnovi opisuju način komunikacije putem fizičkog sloja. Na ovom sloju heterogenost protokola je najprisutnija, posebno kod protokola koji koriste radio valove kao fizički medij. Ta heterogenost se javlja zbog različitih potreba i primjena u sustavima. Tako na ovom sloju imamo različite protokole koji će biti obrađeni u sklopu analize protokola podatkovne poveznice. U nastavku je prikazan komunikacijski modul za protokol ZigBee.



Slika 4.4: ZigBee komunikacijski modul[30]

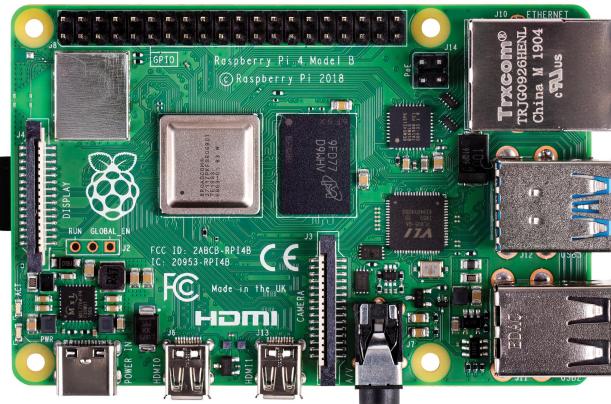
4.2.3. Analiza pristupnih uređaja

4.2.3.1. Raspberry Pi 4 Model B

Raspberry Pi je svojom pojavom na tržištu ponudio malo integrirano računalo za nisku cijenu čime je razvoj Interneta stvari postao pristupačniji za širu zajednicu. Zadnja verzija Pi računala je četvrta s time da su sve tri prijašnje verzije još uvijek u proizvodnji. Razlika između verzija je u procesorskoj snazi, radnoj memoriji, prisutnosti Ethernet i Wi-Fi modula te broju USB i HDMI priključaka. Na Raspberry Pi-u se izvršava GNU/Linux operativni sustav čime se uvelike olakšava razvoj aplikacija i dobiva podrška za postojeća programska rješenja. Na računalu se nalazi 40 priključnih pinova(engl. *GPIO pins*) koji omogućuju spajanje različitih senzorskih pločica, komunikacijskih modula, napajanja i zaslona, čime se postiže visoka modularnost. Raspberry Pi je namijenjen kao stacionaran uređaj zbog njegove relativno visoke potrošnje energije s obzirom na integrirana računala s mikrokontrolerom što zahtjeva konstantan izvor napajanja putem strujnog adaptera. Neke od namjena su kao prilaz za ostale pametne uređaje zbog svoje procesorske snage, kao kontrolni centar sustav za pametan dom korištenjem platformi poput Home Assistanta, a spajanjem modula kamere Raspberry postaje nadzorna kamera.

Raspberry Pi 4 Model B[39] je računalo koje poprima karakteristike stolnog računala. Tako sadrži Broadcom BCM2711, četverojezgreni Cortex-A72 (ARM v8) 64-bit SoC (engl. *System on a chip*), 2, 4 ili 8 gigabajta radne memorije, gigabitni Ethernet priključak, Wi-Fi modul te 2 micro-HDMI priključka koji podržavaju do 4K 60Hz monitore. Te specifikacije dozvoljavaju korištenje RPi4 kao malo stolno računalo, a ne samo kao ugradbeno računalo s nekoliko senzora. Podrška za programske jezike

koji se koriste za razvoj nije ograničena s obzirom na to da uređaj može izvršavati GNU/Linux operativni sustav. Na sljedećoj slici je prikazan Raspberry Pi 4.



Slika 4.5: Raspberry Pi 4 Model B[11]

4.2.3.2. Raspberry Pi Pico

Raspberry Pi Pico[40] je malo integrirano računalo s mikrokontrolerom niske cijene. Pico koristi RP2040 mikrokontroler čime se uvelike smanjuje potrošnja energije, ali i procesorska snaga samog računala čija radna frekvencija iznosi 133MHz s dvije jezgre. Sadrži 264KB SRAM-a te 2MB brze memorije čime je veličina izvršavajućih programa uvelike ograničena. Pico je veličine 21x51mm te tako uz svoju malu potrošnju energije dozvoljava da bude korišten za mobilne potrebe, a sa svojih 26 priključnih pinova dopušta visoku razinu modularnosti. Pristup uređaju se izvodi putem jednog micro-USB 1.1 priključka ili putem UART pinova, ali nema ugrađenih komunikacijskih modula te ako se želi omogućiti pristup mreži potrebno je koristiti zasebne module. Napajanje se provodi putem USB priključka ili putem vanjske baterije spojene na odgovarajuće pinove. Raspberry Pi Pico je ograničen na jednostavne senzorske i aktuatorске primjene zbog svoje ograničene procesorske brzine i kapaciteta pohrane podataka. Podrška za programske jezike koji se koriste za razvoj uređaja je ograničena na MicroPython, CircuitPython, C/C++ te Arduino programski jezik. Uređaj je prikazan na sljedećoj slici.



Slika 4.6: Raspberry Pi Pico[25]

4.2.3.3. Arduino Uno Rev3

Arduino je jedan od najvećih proizvođača malih razvojnih integriranih računala s mikrokontrolerom koji nudi različite vrste računala koja se razlikuju u dimenzijama, priključcima i cijeni. Arduino Uno Rev3[12] je jedno od tih integriranih računala koje je u svojoj prvoj verziji populariziralo i omogućilo jednostavan i jeftin razvoj za sustave Interneta stvari. Shema cijele pločice je sklopovlje otvorenog izvora te je moguće pronaći taj isti uređaj od drugih proizvođača. Treća revizija donosi 8 bitni ATmega328P mikrokontroler koji radi na frekvenciji od 16MHz s 2KB SRAM-a te 32KB programabilne brze memorije. Na pločici se nalazi 20 priključnih pinova od kojih je 14 digitalnih te 6 analognih, uz jedan USB priključak putem kojeg se napaja i jedan zaseban priključak za napajanje. Zbog svoje otvorene i dobro dokumentirane sheme za ovaj uređaj postoji mnoštvo različitih senzorskih pločica i komunikacijskih modula koje je potrebno priključiti ako želimo povezati uređaj na mrežu. Glavna namjena Arduino Una je brz razvoj jednostavnih rješenja Interneta stvari te je zbog pristupačnosti moguće pronaći mnoštvo gotovih projekta kako za senzorska tako i za aktuatora rješenja. Podrška za programske jezike koji se koriste za razvoj uređaja je ograničena na C/C++ te Arduino programski jezik. Na sljedećoj slici je prikazan uređaj.

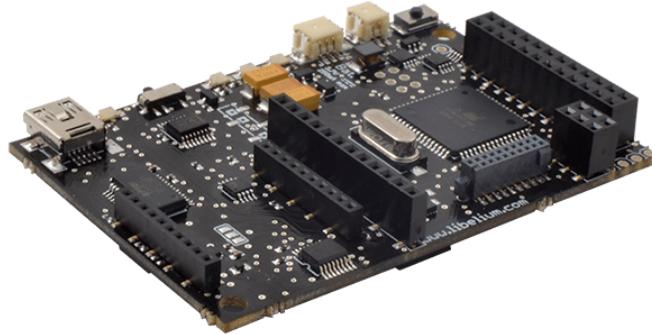


Slika 4.7: Arduino Uno Rev3[12]

4.2.3.4. Libelium Wasp mote

Wasp mote[29] je bežična senzorska platforma otvorenog koda posebno usmjeren na implementaciju načina rada s malom potrošnjom koja omogućuje čvorovima senzora da budu potpuno autonomni s prispojenom baterijom. Životni vijek jednog čvora je od 1 do 5 godina, ovisno o radnom ciklusu i korištenim komunikacijskim modulima. Wasp mote se temelji na modularnoj arhitekturi gdje je ideja integrirati samo module potrebne kako bi se optimizirao rad uređaja. Podržani moduli čine mnoštvo

komunikacijskih, podatkovnih modula te senzorskih pločica. WaspMote sadrži 8-bitni ATmega1281 mikrokontroler koji radi na frekvenciji od 14.74MHz, SRAM od 8KB, 128KB programabilne brze memorije te podršku za SD kratice i mini USB priključak. Jedini programski jezik za razvoj je C++. Na sljedećoj slici je prikazan Libelium WaspMote.



Slika 4.8: Libelium WaspMote[29]

4.2.3.5. ESP32

Razvijen od strane Espressif Systema, ESP32[9] je malo integrirano računalo s mikrokontrolerom niske cijene. ESP32 dolazi s integriranim komunikacijskim modulima za Wi-Fi 802.11 b/g/n i Bluetooth s podrškom za BLE(Bluetooth Low Energy). Na pločici se nalazi dvojezgreni Tensilica Xtensa LX6 mikroprocesor s radnom frekvencijom do 240MHz, SRAM od 512KB, 4MB programabilne memorije, jedan micro USB priključak te 34 priključnih pinova. Napajanje se provodi putem USB priključka ili putem vanjske baterije spojene na odgovarajuće pinove. ESP32 je kod primjene najviše zastupljen kod mobilnih primjena zbog svoje niske potrošnje energije i malih dimenzija. Podrška za programske jezike koji se koriste za razvoj uređaja je ograničena na MicroPython, CircuitPython, Lua, C/C++ te Arduino programski jezik. Prikaz ESP32 računala proizvedenog od strane Joy-IT-a je prikazan na sljedećoj slici.



Slika 4.9: Joy-IT NodeMCU-ESP32[15]

4.2.3.6. Pycom FiPy

Pycom FiPy[17] je malo integrirano računalo temeljeno na ESP32 integriranom računalu. Uz sve prije navedene karakteristike i funkcionalnosti ESP32, FiPy sadrži 8MB programabilne memorije i integrirane komunikacijske module za LoRa, Sigfox i LTE-M. Kao i ESP32 FiPy se koristi za pretežito mobilne primjene zbog velikog broja integriranih bežičnih komunikacijskih modula koji dozvoljavaju velike udaljenosti od baznih prilaza što je posebno izraženo ako se koristi komunikacija putem Sigfoxa čiji domet je do 50 kilometara, LoRa čiji domet je do 40 kilometara i LTE-M čiji domet je do 10 km. Na sljedećoj slici je prikazan Pycom FiPy.



Slika 4.10: Pycom FiPy[17]

4.2.4. Usporedba sigurnosnih mehanizama i primjena

Kako bi komunikacija pristupnih uređaja prema vanjskim poslužiteljima bila sigurna potrebno je korištenje kriptografskih algoritama. Ti algoritmi zahtijevaju brzu procesnu snagu što nije slučaj za sve prije navedene uređaje osim Raspberry Pi 4 Model B koji ima četverojezgreni procesor radne frekvencije od 1.5GHz. Zato se kao ispomoć u kriptografskim postupcima na pločicama također nalazi i zasebni koprocesor za izvođenje kriptografskih algoritama ili podrška za sklopovsko ubrzanje određenih kriptografskih algoritama. Tako ESP32 i Pycom FiPy nude podršku za sklopovsko ubrzanje algoritama SHA, RSA i AES uz generator nasumičnih brojeva, Libelium Wasp mote nudi podršku za kriptografske biblioteke za algoritme AES, RSA, MD5 i SHA iako nije navedena podrška za sklopovsko ubrzanje ili koprocesor. Arduino Uno Rev3 ne nudi nikakvu posebnu podršku za kriptografske algoritme iako neke verzije drugih proizvođača nude podršku za koprocesor za kriptografske algoritme. To je i slučaj s Raspberry Pi Picom gdje drugi uređaji temeljeni na Pico RP2040 mikroprocesoru imaju dodatan koprocesor na pločici. Raspberry Pi 4 Model B se približava poziciji stolnog računala te zbog svoje procesorske snage nema potrebe za dodatnim koprocesorima i na njemu se mogu izvoditi svi kriptografski postupci. Zato ako postoji potreba da uređaj samostalno obavlja zadatke i potrebno je osigurati komunikaciju između tog

uređaja i bazne stanice ili poslužitelja najbolje je koristiti uređaje: Raspberry Pi 4 Model B, ESP32 i Pycom FiPy, dok Raspberry Pi Pico i Arduino Uno Rev3 je najbolje koristiti kao integrirano senzorsko računalo koje je dodatno putem priključnih pinova ili USB-a povezano sa snažnijim uređajem.

Primjena svih navedenih uređaja ovisi o njihovim izvorima napajanja, veličini, podršci za komunikacijske module i procesorskoj snazi. Ako gledamo procesorsku snagu i izvor napajanja tu dolazimo do pitanja energetske učinkovitosti uređaja koja ograničava mogućnost upotrebe baterija kao izvora napajanja. Tako će se Raspberry Pi 4 Model B koristiti kao stacionaran pametan uređaj s vanjskim napajanjem, dok ostali navedeni uređaji imaju mogućnost biti baterijski napajani što dozvoljava mobilnu primjenu. Zbog svoje velike procesorske snage Raspberry Pi 4 Model B može djelovati kao poslužitelj i prilaz za ostale pametne uređaje dok su ostali uređaji većinom ograničeni na jednostavne senzorske i aktuatorске primjene. Svi navedeni uređaji imaju podršku za vanjske komunikacijske module te nisu ograničeni na tom području iako neki od uređaja dolaze s već integriranim modulima što dozvoljava brz, jednostavan i jeftin razvoj.

4.3. Fizički sloj i sloj podatkovne poveznice

Fizički sloj se definira fizičkim medijem kojim se prenose podaci. Postoje dvije vrste kategorije medija, žičani i bežični, na kojima se zatim temelje protokoli podatkovne poveznice. Uloga fizičkog sloja je da definiraju fizički medij koji se koristi u komunikaciji, svojstva tog medija poput frekvencije koja će se koristiti u radiovalnoj komunikaciji, sinkronizaciju, pakete i linijske kodove dok sloj podatkovne poveznice definira strukturu paketa, automate stanja, adresiranje, sinkronizaciju, kontrolu protoka i korekciju grešaka. Na ovim slojevima se nalazi velika heterogenost protokola zbog različitih potreba u sustavima Interneta stvari. Heterogenost protokola je najizraženija u bežičnoj komunikaciji zbog različitih potreba u dometu komunikacije, brzini prijenosa informacija, latenciji i mobilnosti uređaja. Sve ove potrebe su posebno izražene jer upotrebom bežične komunikacije proširujemo mogućnosti primjena uređaja, dok žičane komunikacije uglavnom ograničavamo za potrebe komunikacije prilaza i vanjskih servisa zbog stabilne i brze komunikacije. U nastavku se analiziraju i uspoređuju samo neki od protokola fizičkog sloja i sloja podatkovne poveznice temeljenih na bežičnim tehnologijama.

4.3.1. Analiza protokola

4.3.1.1. IEEE 802.11 Wi-Fi

IEEE 802.11 Wi-Fi[1] je porodica bežičnih komunikacijskih protokola temeljenih na radiofrekvencijskom prijenosu. Varijante protokola Wi-Fi su jedni od najbolje zastupljenih komunikacijskih protokola zbog svoje pojave na tržištu s uređajima kojima je cilj bio mobilna uporaba poput prijenosnih računala i mobilnih telefona. Kroz godine se počela pojavljivati potreba za bržim prijenosom podataka, manjom latencijom i mogućnošću povezivanja više uređaja na jednu baznu stanicu (engl. *access point*) što je dovelo i do razvoja različitih standarda koji svi čine jednu porodicu protokola. U tablici su prikazani različiti standardi, frekvencije koje koriste i maksimalna propusnost koja se može postići.

Tablica 4.1: Varijante protokola Wi-Fi

Standard	Ime	Frekvencije	Maksimalna propusnost
802.11a	Wi-Fi 2	5 GHz	54 Mbps
802.11b	Wi-Fi 1	2.4 GHz	11 Mbps
802.11g	Wi-Fi 3	2.4 GHz	54 Mbps
802.11n	Wi-Fi 4	2.4 i 5 GHz	600 Mbps
802.11ac	Wi-Fi 5	2.4 u 5 GHz	1.3 Gbps
802.11ax	Wi-Fi 6	2.4 i 5 GHz	10-12 Gbps
802.11ah	Wi-Fi HaLow	EU: 866 MHz	347 Mbps

IEEE 802.11i je dopuna na IEEE 802.11 *Wired Equivalent Privacy* (WEP) sigurnosni mehanizam definiran u originalnom 802.11 standardu koji je zbog svojih slabih sigurnosnih svojstva neadekvatan standard. IEEE 802.11i također nosi naziv i *Wi-Fi Protected Access II* (WPA2). WPA2 je trenutno najkorišteniji za sigurnost komunikacija vezanih uz protokole Wi-Fi. Prethodni WPA je također uz WEP prestao biti korišten u široj upotrebi zbog pronađenih sigurnosnih propusta vezanih uz slabe kriptografske algoritme korištene u šifriranju prometa. WPA3 je zadnji sigurnosni standard koji nudi poboljšanja u autentifikacijskom postupku i šifriranju prometa te će biti zadani standard za Wi-Fi 6 komunikaciju. U nastavku je prikazana tablica koja prikazuje razlike u svim sigurnosnim mehanizmima. Zbog raširenosti u nastavku će se promatrati sigurnosni mehanizmi WPA2.

Tablica 4.2: Usporedba sigurnosnih Wi-Fi standarda

	WEP	WPA	WPA2	WPA3
Razvijen godine	1997.	2003.	2004.	2018.
Razina sigurnosti	Jako niska	Niska	Visoka	Jako visoka
Šifriranje	RC4	TKIP	AES-CCMP AES-GCMP	
Veličina ključa	64 bita 128 bita	64 bita 128 bita	128 bita	128 bita 256 bita
Autentifikacija	Otvoreni pristup Dijeljeni ključ	PSK AAAK	PSK AAAK	AES-CCMP AES-GCMP
Integritet	CRC-32	64 bitni MIC	AES-CCMP	SHA-2

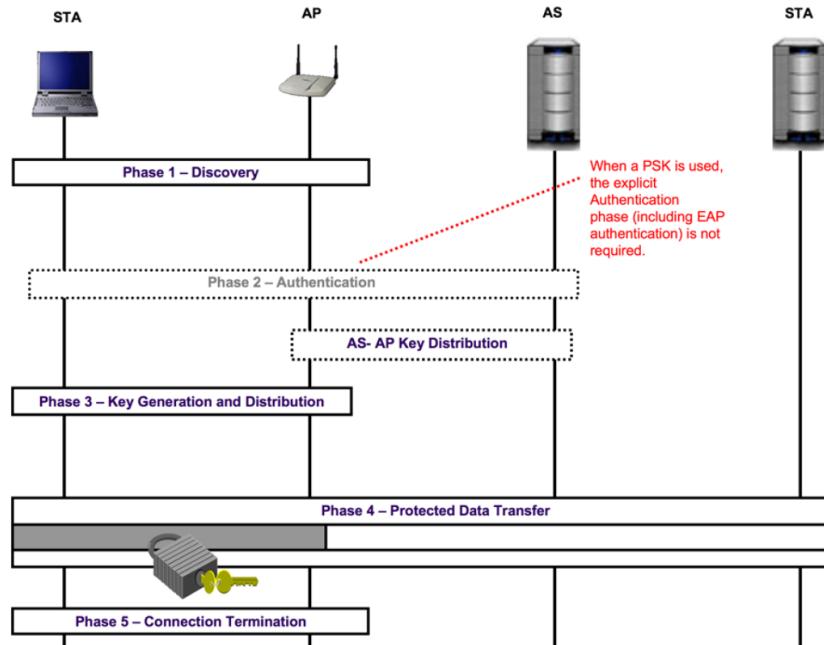
WPA2 uvodi koncept *Robust Security Network* (RSN), koji se definira kao bežična sigurnosna mreža koja omogućuje kreiranje sigurnih bežičnih konekcija naziva *Robust Security Network Associations* (RSNA). RSNA omogućava sigurnosne mehanizme napredne autentifikacije korisnika, upravljanje kriptografskim algoritmima, povjerljivost podataka, autentifikacija i integritet izvora podataka te zaštitu od napada ponavljanjem. RSNA koristi dvije autentifikacijske metode koje se koriste kako bi se odvijalo izvođenje kriptografskih ključeva za autentifikacija i integritet podataka. *Pre-Shared Key* (PSK) se koristi u nekomercijalnim mrežama gdje svi korisnici koji pristupaju mreži koriste jedan ključ, tj. lozinku. *Authentication, Authorization, and Accounting Key* (AAAK) koji se pristupnoj točki dostavlja putem *Extensible Authentication Protocol* (EAP) koji omogućuje autentifikaciju korisnika putem identiteta gdje svaki korisnik ima svoje pristupne podatke i koristi se većinom u komercijalnim mrežama.

Tri različite komponente RSNA su: krajnji uređaji koji pristupaju mreži (STA) (engl. *stations*), pristupna točka (AP) (engl. *access point*) i autentifikacijski poslužitelji (AS). U nastavku se promatraju faze za uspostavu i raskid veze, a na slici 4.11 su prikazane faze korištenjem PSK autentifikacijske metode.

- 1. Otkrivanje:** STA identificira AP za mrežu s kojom želi uspostaviti komunikaciju. STA locira AP primanjem jednog od AP-ovih periodičnih *Beacon* okvira ili slanjem *Probe* zahtjeva. Nakon što STA identificira AP, dolazi do izmjenjivanja

nja okvira za pregovor oko komunikacijskih parametara. Na kraju prve faze su uspostavljeni kriptografski protokoli za povjerljivost i integritet podataka, autentifikacijska metoda i način distribucije ključeva.

2. **Autentifikacija:** U slučaju metode korištenjem AAAK u ovoj fazi se izvodi međusobna autentifikacija STA i AP-a. Autentifikacijski okviri prolaze kroz AP do AS-a kako bi se korištenjem EAP-a mogla odviti autentifikacija. AP blokira sve okvire STA koji nisu autentifikacijski dok se ne izvrši autentifikacija STA. Kada se autentifikacija izvrši AAAK se spremi na STA i AP. Metoda korištenjem PSK nema ovu fazu zbog korištenja jednog ključa za pristup.
3. **Generiranje i distribucija ključeva:** Tijekom ove faze AP i STA izvode nekoliko operacija koje generiraju kriptografske algoritme za daljnju komunikaciju. To se izvodi kroz dvije vrste rukovanja: četverostruko rukovanje i rukovanje grupnim ključevima.
4. **Zaštićeni prijenos podataka:** STA i AP izmjenjuje podatke sigurnim putem korištenjem sigurnosnih politika i kriptografskih ključeva uspostavljenih tijekom prve tri faze. Zato što se sigurni prijenos odvija samo između STA i AP-a, potrebno je zasebno zaštiti podatke tijekom ostatka prijenosa.
5. **Raskid konekcije:** STA i AP prekidaju svoju sigurnu vezu i brišu podatke o svojoj međusobnoj povezanosti[32].



Slika 4.11: Faze uspostava i raskida veze korištenjem PSK metode[32]

4.3.1.2. Bluetooth Low Energy

Bluetooth Low Energy (BLE)[13] je bežični protokol male snage te je jedan od dva protokola pod Bluetooth nazivom. Drugi protokol je Bluetooth Classic koji se koristi za kontinuiranu komunikaciju velikog skupa podataka na maloj udaljenosti te kao takav je korišten u Internetu stvari, ali za potrebe stacionarnih uređaja sa stalnim izvorom napajanja koji kontinuirano šalju podatke. BLE koristi periodični prijenos manjih podatkovnih paketa te u trenucima kada ne sudjeluje u razmjeni podataka je u stanju mirovanja što uvelike doprinosi smanjenju potrošnje energije. Tako se omogućuje razvoj manjih mobilnih uređaja koji mogu biti pokretani duže vrijeme na maloj bateriji.

Na fizičkom sloju BLE koristi ISM (engl. *Industrial, Scientific, and Medical*) radiofrekvencijski pojas na frekvenciji od 2.4GHz. Brzina podatkovnog prijenosa je 1Mbps za verzije 4.2 ili starije, a za verzije 5 i novije može postići do 2Mbps. Udaljenost na kojoj uređaji mogu komunicirati ovisi o preprekama koje se nalaze, vrsti antene, kućištu i orientaciji uređaja, ali teoretska maksimalna udaljenost je između 100 i 400 metara.

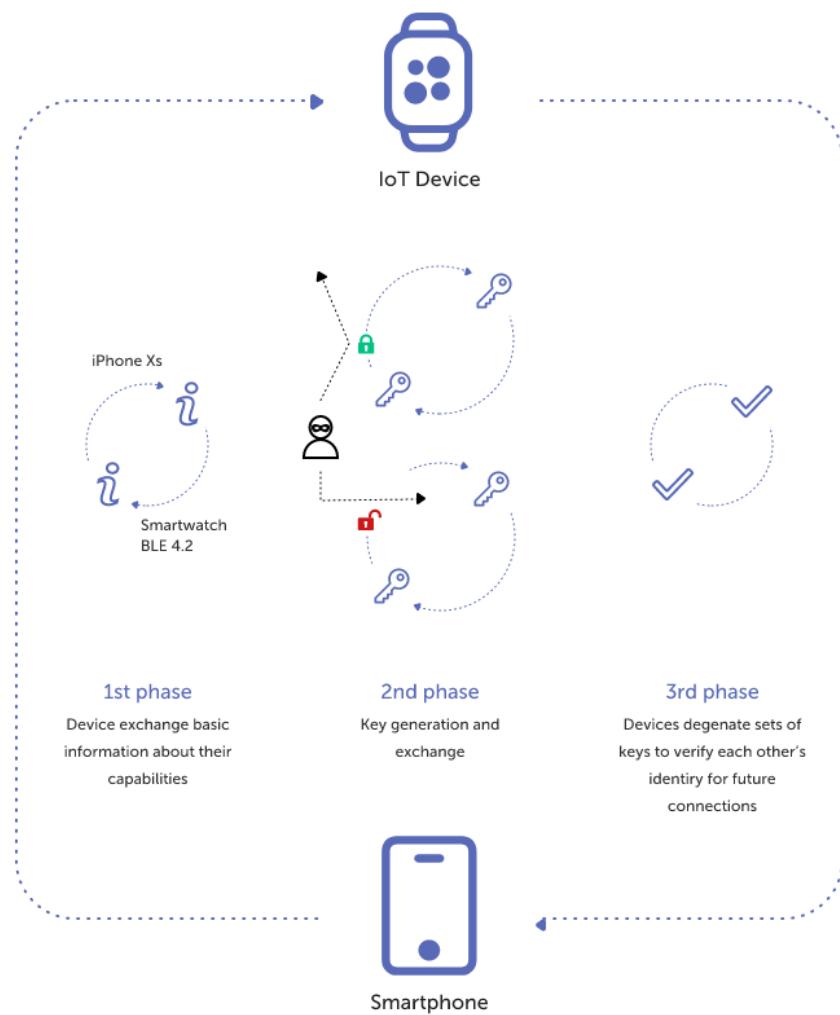
Topologija BLE mreže podrazumijeva dvije vrste uređaja:

Centralni uređaji: imaju veću procesnu snagu i odgovorni su za kontroliranje perifernih uređaja. Na njima se izvode aplikacije posebno izrađene za interakciju s perifernim uređajima. Neki od primjera centralnih uređaja su: pametni telefoni, tableti i računala.

Periferni uređaji: mala integrirana računala koja obnašaju zadaće opažanja koristeći senzore te prikupljene podatke šalju na centralne uređaje za obradu. Neki od primjera perifernih uređaja su: pametni satovi, nosivi medicinski uređaji i industrijski termostati.

Kako bi se mogla odvijati komunikacija između centralnih i perifernih uređaja prvo je potrebno uspostaviti konekciju postupkom koji se naziva uparivanje (engl. *pairing*). Uparivanje se odvija u dvije ili tri faze, tijekom koji uređaji izmjenjuju podatke potrebne za autentifikaciju, poput ključeva i ulazno/izlaznih sposobnosti. Treća faza je neobavezna te se dešava u slučaju da će uređaji uspostaviti vezu (engl. *bond*). Vezivanje (engl. *bonding*) je proces u kojem uređaji spremaju autentifikacijske podatke koje su izmijenili tijekom prvobitnog uparivanja kako bi se uklonila potreba za ponovnim uparivanje tijekom ponovne uspostave komunikacije. Tijekom prve faze uređaji razmjenjuju informacije o identitetu i sposobnostima uređaja i komunikacija nije šifrirana. Druga faza je posvećena generiranje i izmjeni ključeva te je tijekom ove faze moguć

napad *man-in-the-middle* napad gdje napadač može preuzeti kontrolu nad uređajem i podacima koje se prenose. Kod treće faze uparivanja uređaji generiraju set ključeva za autentifikaciju kod budućih uspostava veza. Ti ključevi mogu biti par potpisnih ključeva za razlučivanje konekcije (engl. *Connection Signature Resolving Keys*), koji se koriste za digitalni potpis podataka, te ključevi za razlučivanje identiteta (engl. *Identity Resolving Keys*), koji se koriste za generiranje privatnih MAC adresa i pretraživanje. Kriptografski algoritam koji se koristi za šifriranje prometa je AES (engl. *Advanced Encryption Standard*) sa 128 bitnim ključem izведен tijekom postupka uparivanja. Prikaz postupka uparivanja je prikazan na sljedećoj slici.



Slika 4.12: Uparivanje uređaja protokolom BLE [36]

BLE podržava dvije vrste konekcija *Legacy* i *Secure*. Legacy konekcije mogu biti implementirane za verzije 4.0, 4.1 i 4.2. Uređaji razmjenjuju vrijednost privremenog ključa (engl. *Temporary Key*) i koriste tu vrijednost kako bi generirali kratkoročni ključ (engl. *Short Term Key*) koji se zatim koristi za autorizaciju konekcije. BLE Legacy ko-

nekcije su nesigurne, ali mogu biti osigurane s određenim metodama uparivanja. Secure konekcije su predstavljane s verzijom 4.2 i nisu kompatibilne sa starijim verzijama protokola. Ove konekcije implementiraju algoritam eliptičke krivulje Diffie-Hellman (ECDH) za generiranje ključeva i donose kompleksniji proces autentifikacije ključima. Tako se sprječava napad prisluškivanjem i uređaji mogu biti dodatno osigurani s određenim metodama uparivanja opisanim u nastavku.

Just Works metoda uparivanja je zadana metoda za većinu BLE mreža. Kod Legacy konekcija vrijednost privremenog ključa tijekom druge faze uparivanja je postavljena na 0 i uređaji generiraju kratkoročni ključ na temelju te vrijednosti. Takva vrsta sparivanja je jako nesigurna i ne nudi nikakvu vrstu zaštite, već samo način za uspostavu veze. Ova vrsta uparivanja može biti kompleksna i pouzdana za osnovnu sigurnost konekcije kroz sljedeće korake:

1. Korištenjem kriptografskog algoritma ECDH, oba uređaja generiraju privatni i javni par ključeva i izmjenjuju javni ključ.
2. Uređaj koji prima zahtjev za konekcijom generira nasumičnu vrijednost i koristi tu vrijednost kako bi generirao potvrdu vrijednost. Uređaj šalje obje vrijednosti uređaju koji je inicirao konekciju.
3. Uređaj koji inicira konekciju generira vlastitu potvrdu vrijednost s nasumičnom vrijednosti koju je primio od drugog uređaja. Uspoređuje vlastitu potvrdu vrijednost s potvrdom vrijednosti koju je dobio od drugog uređaja.
4. Ako su vrijednosti identične, uređaji uspostavljaju vezu.

Ova metoda nudi zaštitu od pasivnog prisluškivanja, ali ne i od *man-in-the-middle* napada te se ne preporučuje za primjene gdje se razmjenjuju osjetljivi podaci.

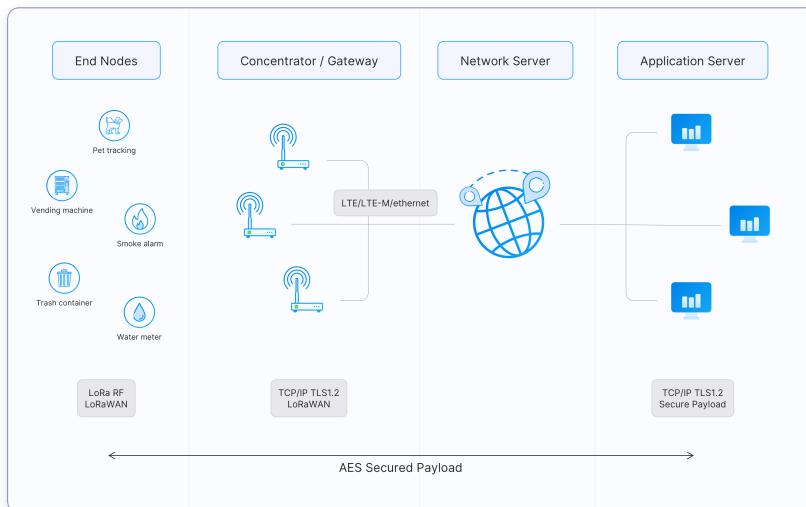
Out of Band metoda uparivanja dozvoljava da se određeni podaci tijekom druge faze uparivanja šalju putem nekog drugog bežičnog protokola kako se ključevi ne bi razmjenjivali putem manje sigurnog protokola BLE ili kada uređaji razmjenjuju osjetljive podatke. Krajnja sigurnost ove metode ovisi o protokolu koji se koristi te se na taj način može sprječiti napadi pasivnim prisluškivanjem i *man-in-the-middle* napadi.

Passkey metoda uparivanja koristi korisnike kao dio procesa. Postoje različiti načini implementacije ove metode. Jedan od primjera je da uređaj generira lozinku od šest znamenki, koju tada korisnik treba unijeti na drugom uređaju. Na ovaj način se svaki uređaj ručno verificira od strane korisnika što zahtjeva da uređaji imaju neki oblik unošenja i prikazivanja podataka. Ovim načinom se sprječavaju napadi pasivnim prisluškivanjem i *man-in-the-middle* napadi.

Metoda uparivanja numeričkom usporedbom je dostupna samo za Secure konekcije te također zahtjeva sudjelovanje korisnika. Nakon druge faze uparivanja uređaji koriste nasumične vrijednosti koje su prije razmijenili kako bi generirali šestoznamenkastu vrijednost koju korisnik treba usporediti na oba uređaja. Ova metoda nudi zaštitu od *man-in-the-middle* napada jer nijedan uređaj se ne može povezati na mrežu bez ručne verifikacije[36].

4.3.1.3. LoRaWAN

LoRaWAN (engl. *Long Range Wide Area Network*)[20] je standard za bežičnu komunikaciju koji dozvoljava komunikaciju uređaja Interneta stvari preko velikih udaljenosti s minimalnom energetskom potrošnjom. LoRa i LoRaWAN su pojmovi koji se koriste naizmjenično iako definiraju različite slojeve Internet stvari složaja. LoRa definira fizički sloj dok LoRaWAN definira protokol i arhitekturu sustava. LoRa definira različite radiofrekvencijske pojase koji pripadaju ISM pojasevima. U Europi LoRa koristi frekvencije na pojasu oko 869MHz, dok u Sjevernoj Americi koristi 923MHz čime se javlja nekompatibilnost uređaja proizvedenih za ta dva tržišta. Domet komunikacije ovisi o mnogo parametara poput naseljenosti područja u kojem se odvija komunikacija. Za urbana područja domet je oko 5km, predgrađa 10km, a u nenaseljenim područjima može dostići i do 80km. LoRaWAN arhitektura se sastoji od: krajnjih čvorova (uređaja), LoRa prilaza, mrežnog poslužitelja i krajnjih aplikacija. Prikaz topologije LoRaWAN arhitekture je na sljedećoj slici.



Slika 4.13: Topologija LoRaWAN arhitekture[21]

Krajnji čvorovi mogu biti senzori ili aktuatori, često su napajani baterijama i imaju

visoku razinu energetske učinkovitosti te tako mogu dugo funkcionirati na jednom punjenju. Postoje 3 kategorije čvorova. Klasa A je najučinkovitija te može primati poruke samo nakon uspješnog slanja poruke s uređaja. Klasa B nadograđuje klasu A i dodaje mogućnost primanja poruka u predodređenim intervalima. Klasa C može primati podatke samo kada ih ne šalje. Uređaji na mreži koriste sljedeće vrste adresa i identifikatora:

DevEUI: Jedinstveni identifikator formata EUI64 (engl. *Extended Unique Identifier*) adrese veličine 64 bita. Usporedivo s MAC adresama za TCP/IP uređaje.

DevAddr: Adresa uređaja veličine 32 bita dodijeljena ili odabrana na mreži. Usporedivo s IP adresama za TCP/IP uređaje.

AppEUI: Aplikacijski identifikator formata EUI64 adrese. Jedinstveno identificira aplikacijskog davatelja usluga uređaja. AppEUI se spremi na krajnji čvor prije nego se izvodi aktivacijski postupak za verzije prije 1.1.

JoinEUI: Identifikator pristupnog poslužitelja formata EUI64 adrese. Jedinstveno identificira pristupnog poslužitelja. JoinEUI se spremi na krajnji čvor prije nego se izvodi aktivacijski postupak za verziju 1.1.

Fport: Identificira vrata aplikacijskog procesa. Usporedivo sa TCP/UDP vratima za TCP/IP uređaje.

LoRa prilazi primaju LoRa poruke od krajnjih uređaja te samo prosljeđuju te podatke na mrežne poslužitelje. Prilazi su povezani na Internet putem drugih protokola poput Wi-Fi-a, Etherneta ili mobilnih mreža.

Mrežni poslužitelji upravljaju cijelom LoRaWAN mrežom. Zadaće poslužitelja su:

- agregiranje svih dolaznih podataka s LoRa prilaza,
- usmjeravanje poruka na aplikacijske poslužitelje,
- konfiguraciju prilaza,
- optimiziranje brzine prijenosa podataka, vremena emitiranja i potrošnje energije na mreži,
- aktivaciju krajnjih uređaja,
- slanje potvrda o primljenim porukama.

Pristupni poslužitelj obrađuju pristupne zahtjeve krajnjih uređaja. Poslužitelj obavlja funkcije spremanja root kriptografskih ključeva, generira ključeve sjednice i prijenosi ključeve sjednice mrežnim i aplikacijskim poslužiteljima. Pristupni poslužitelj je uveden u LoRaWAN mrežu od verzije 1.0.4 i 1.1.

LoRaWAN 1.1 nudi nekoliko razina sigurnosti. Mrežni poslužitelj ne treba imati informaciju o sadržaju poruke koja se usmjeruje, a da nije relevantna za mrežu ili infrastrukturu. Zato postoje dva različita kriptografska 128 bitna ključa koji koriste algoritam za šifriranje AES i 2 ključa za provjeru integriteta poruka kod uređaja kako bi se postigla zaštita podataka na dvije razine. Ključevi korišteni nakon povezivanja uređaja na mrežu su sljedeći:

SNwkSIntKey: Ovo je ključ mrežne sjednice koji koristi uređaj za potvrdu integriteta svake primljene poruke putem *Message Integrity Code* (MIC) provjere i za izračun MIC-a za pola odlazne poruke.

FNwkSIntKey: Ovo je ključ mrežne sjednice koji koristi uređaj za izračun MIC-a cijelog ili dijela odlazne poruke.

NwkSEncKey: Ovo je ključ mrežne sjednice koji koristi uređaj za šifriranje i dešifriranje dolaznih i odlaznih poruka MAC naredbi.

AppSKey: Ovo je ključ aplikacijske sjednice koji koriste aplikacija i uređaj za šifriranje i dešifriranje poruka koje izmjenjuju.

Mrežni ključ sjednice (NwkSEncKey) se koristi za komunikaciju između krajnjeg čvora i mrežnog poslužitelja. Ključ potvrđuje integritet svake poruke putem *Message Integrity Code* (MIC) provjere. Ta provjera je slična provjeri koristeći kontrolnu sumu uz razliku da onemogućuje neautorizirane promjene sadržaja poruke. Ova provjera se također koristi za mapiranje DevAddr na jedinstvene DevEUI i AppEUI. NwkSEncKey se nakon aktivacije čvora dijeli s mrežom.

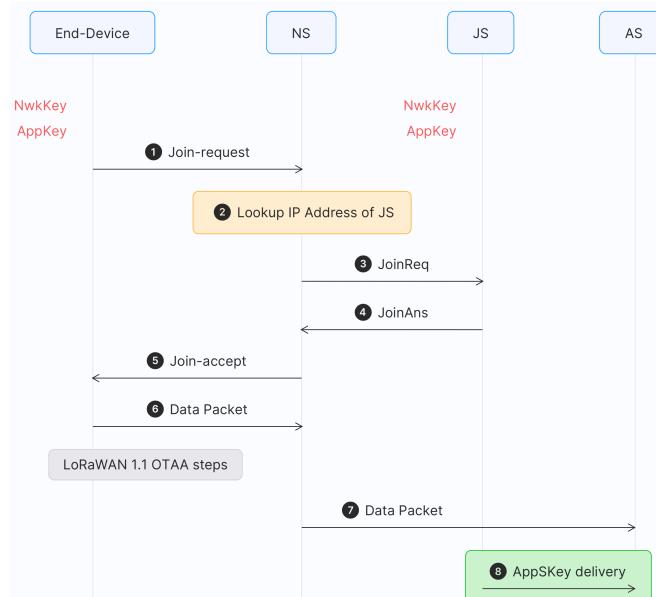
Aplikacijski ključ sjednice (AppSKey) se koristi za šifriranje i dešifriranje sadržaja poruka. Sadržaj poruke je šifriran cijelim putem između čvora i aplikacijskog poslužitelja. Ovaj ključ se generira kao i NwkSkey nakon aktivacije čvora, ali samo čvor i aplikacijski poslužitelj imaju taj ključ, što omogućava potpunu tajnost podataka dok ne stigne do aplikacije.

Za aktivaciju čvora i pristupanje mreži kako bi se mogla odvijati komunikacija i kako bi se stvorili kriptografski ključevi se može odviti s dva različita postupka: ABP (engl. *Activation By Personalization*) i OTAA (engl. *Over The Air Activation*).

ABP je najjednostavniji način za aktivaciju i pristup LoRa uređaju na mrežu. Dobavljač uređaja od drevatelja mrežne usluge otkupi određeni raspon DevAddr adresa i opskrbi svoje uređaje s tim adresama. Nakon što se uređaj upali odmah je spreman za komunikaciju s prilazima. Koraci u postupku aktivacije su:

1. Dobavljač uređaja otkupi mogućnost povezivanja od drevatelja mrežnih usluga. AppSKey je generiran unaprijed.
2. Drevatelj mrežnih uređaja dostavlja NwkSkey i DevAddr za svaki DevEUI.
3. Prije uporabe uređaja se opskrbuje uređaje s: NwkSkey, AppSKey i DevAddr
4. Kod prvog korištenja uređaja, nema nikakvog koraka za aktivacijom i pristupa uređaju mreži jer uređaji već mogu komunicirati s mrežom jer imaju sve potrebne informacije.

OTAA je za razliku od ABP postupka komplikiraniji, ali dopušta pristup bilo kojoj mreži u blizini i ne zahtijeva poseban dogovor između dobavljača uređaja i drevatelja mrežnih usluga. Postupak je različit za verziju 1.1 i ranije verzije, a ovdje će se gledati postupak aktivacije za verziju 1.1. Kako bi se sigurnim putem dobavili potrebni ključevi i adresa uređaja postupak koristi dva dodatna kriptografska ključa naziva AppKey i NwkKey koji uz DevEUI trebaju biti spremjeni na uređaju prije postupka aktivacije. To su kriptografski ključevi od 128 bita korišteni za kriptografski algoritam AES. Na pristupnom poslužitelju su spremjeni AppKey, NwkKey i DevEUI. Koraci u postupku aktivacije su prikazani na sljedećoj slici i opisani u nastavku.



Slika 4.14: Tijek LoRaWAN 1.1 OTAA postupka[19]

1. LoRa uređaj šalje Join-request zahtjev potpisani s NwkKey. Zahtjev za pristup mreži sadrži JoinEUI, DevEUI i DevNonce. DevNonce je dvobitni brojač koji počinje s vrijednošću 0 i inkrementira se sa svakim Join-request zahtjevom, a koristi se za sprječavanje napada ponavljanjem. Ovaj zahtjev nije šifriran. Zahtjev LoRa prilazi prosljeđuju na mrežni poslužitelj.
2. Mrežni poslužitelj koristi DNS kako bi doznao IP adresu pristupnog poslužitelja iz JoinEUI polja.
3. Mrežni poslužitelj šalje JoinReq zahtjev na pristupni poslužitelj. Zahtjev sadrži Join-request zahtjev, MAC verziju, DevAddr i druge servisne informacije.
4. Pristupni poslužitelj obrađuje Join-request zahtjev i šalje JoinAns poruku mrežnom poslužitelju. Poruka sadrži Join-accept poruku, NwkSkey, *Serving Network session integrity key* (SNwkSIntKey), *Forwarding Network session integrity key* (FNwkSIntKey), *Network session encryption key* (NwkSEncKey) i šifrirani AppSKey.
5. Mrežni poslužitelj prosljeđuje primljenu Join-accept poruku krajnjem uređaju. Uređaj izračunava MIC i generira mrežne ključeve sesije i AppSKey. FNwkSIntKey, SNwkSIntKey i NwkSEncKey se izvode iz NwkKey.
6. Mrežni poslužitelj prima odgovor od uređaja te šalje DevEUI i šifriran AppSKey s porukom aplikacijskom poslužitelju.
7. Aplikacijski poslužitelj prima šifriran AppSKey s porukom i dešifrira AppSKey koristeći tajni ključ izmijenjen između pristupnog i aplikacijskog poslužitelja. Tada koristi AppSKey za dešifriranje poruke. Ako šifrirani AppSKey nije primljen tada se dovija još jedan korak.
8. Aplikacijski poslužitelj zahtjeva AppSKey izravno od pristupnog poslužitelja slanjem AppSKeyReq zahjeva koji sadrži DevEUI uređaja. AppSKey je šifriran koristeći zajedničku tajnu pristupnog i mrežnog poslužitelja. Pristupni poslužitelj šalje šifrirani AppSKey i DevEUI aplikacijskom poslužitelju u AppSKeyAns poruci. Aplikacijski poslužitelj dešifrira AppSKey i pomoću njega dešifrira poruku uređaja.
9. Nakon aktivacije uređaj ima spremlijen DevAddr, FNwkSIntKey, SNwkSIntKey, NwkSEncKey i AppSKey[19].

LoRaWAN mreže se u načinu pristupa i potrebne implementacije dijele u tri skupine: komercijalne, gdje mrežu pružaju operateri i naplaćuju pristup, javne, u koje se može bilo tko uključiti i pružati pristup te privatne u kojima svatko može pokrenuti svoju vlastitu privatnu mrežu.

4.3.2. Usporedba sigurnosnih mehanizama i primjena

Protokoli navedeni u analizi koriste bežičnu komunikaciju između krajnjih uređaja i bazne stanice ili prilaza. Važno je zaštititi podatke koji se prenose u komunikaciji kako bi se spriječio napad pasivnim prisluškivanjem. Pasivno prisluškivanje se zbog bežične prirode komunikacije može lako izvesti nad nešifriranim podacima. LoRaWAN i BLE koriste algoritam AES sa 128 bitnim ključem koji nudi visoku razinu zaštite. Wi-Fi nudi nekoliko različitih kriptografskih algoritama ovisno o korištenom sigurnosnom standardu. WEP koristi algoritam RC4 sa 64 ili 128 bitnim kriptografskim ključem, WPA koristi algoritam TKIP sa 64 ili 128 bitnim kriptografskim ključem, WPA2 koristi algoritam AES sa 128 bitnim ključem, dok WPA3 koristi AES sa 128 ili 256 bitnim ključem ovisno o načinu rada algoritma AES. *Man-in-the-middle* napadi su također prisutni u bežičnoj komunikaciji u kojoj se napadač može naći kao posrednik u komunikaciji. Kako bi se spriječila ta vrsta potrebna je sigurna implementacija postupka pristupa mreži. BLE nudi metode poput *Passkey*, *Out of Band* i uparivanje numeričkom usporedbom koji sprječavaju tu vrstu napada. LoRaWAN 1.1 donosi unaprjeđenja nad prijašnjim verzijama korištenjem ABP i OTAA metoda. Wi-Fi sigurnosni standardi WEP, WPA i WPA2 imaju otkrivene sigurnosne propuste koji omogućuju tu vrstu napada, dok WPA3 donosi sigurnosne mehanizme koji to sprječavaju.

Wi-Fi je najpristupačniji protokol za razvoj Internet stvari rješenja zbog svoje raširenosti pristupnih točaka koje se koriste u različitim prostorima poput domova, ureda i javnim prostorima. Domet komunikacije ovisi o varijanti protokola, ali većinom se kategoriziraju kao protokol srednje udaljenosti uz Wi-Fi HaLow koji dozvoljava veće udaljenosti do 1km. BLE je drugi najpristupačniji protokol za razvoj također zbog svoje raširenosti u mobilnim telefonima i osobnim računalima. Primjena protokola je zbog svog kratkog dometa najprisutnija u rješenjima koje se nalaze na ljudskom tijelu poput medicinskih i *fitness* uređaja koji se lako povezuju s mobilnim telefonima. LoRaWAN je protokol najvećeg dometa te se najviše koristi u senzorskim i aktuatorским primjenama poput senzora požara i brojila. Uređaji većinom dolaze s podrškom za Wi-Fi i BLE komunikacijske module dok LoRaWAN zahtjeva zaseban modul što uz

neraširenost protokola često nije pristupačno za razvoj.

4.4. Mrežni sloj

Mrežni sloj u složaju Interneta stvari omogućava prijenos informacija od prilaza do servisa koji se nalaze u vanjskoj mreži. Glavni način za komunikaciju između mreža je Internet protokol(IP).

4.4.1. Analiza protokola

4.4.1.1. IP

IP definira adresu uređaja kako bi paketi koji se šalju mogli biti usmjereni na taj uređaj. IP se dijeli na protokole IPv4 i IPv6. Ta podjela postoji zbog ograničenosti IPv4 na 32 bitne adrese koje dozvoljavaju oko 4.2 bilijuna (2^{32}) adresa od kojih su neke rezervirane za posebne potrebe. IPv4 je postao usko grlo na mrežnom sloju, posebice razvitkom Interneta stvari te sve većem broju umreženih mobilnih uređaja. IPv6 donosi 128 bitne adrese koje dozvoljavaju oko $3.4 * 10^{38}$ adresa. Obje verzije protokola sadrže i proširenje pod nazivom IPSec (engl. *Internet Protocol Security*) koji vrši autentifikaciju i šifriranje paketa kako bi se osigurala komunikacija između dva računala preko različitih mreža. Glavna namjena ovog protokola je za uspostavu virtualnih privatnih mreža(VPN) te kao takav je rijetko u uporabi u Internet stvari sustavima već se autentifikacija i šifriranje podataka vrši na aplikacijskom sloju i sloju podatkovne poveznice te neće biti obrađen u sklopu ovog rada.

4.4.2. Usporedba sigurnosnih mehanizama i primjena

IP ne donosi nikakve sigurnosne mehanizme osim kontrolne sume kod IPv4. Razlog nepostojanja dodatnih mehanizama je postojanje podrške za šifriranjem na slojevima podatkovne poveznice i aplikacijskog sloja te postojanje kontrolne sume na tim slojevima uključujući i transportni sloj. Iako IPv4 sadrži kontrolnu sumu, ta opcija u zaglavlju je izostala iz IPv6 kako bi se smanjila potrebno vrijeme za procesiranjem s obzirom na to da protokoli viših i nižih slojeva već sadrže tu mogućnost. Nepostojanje sigurnosnih mehanizama kod ovih protokola je najizraženije u takozvanim *IP spoofing* napadima. Taj napad se izvodi promjenom izvorišne adrese kako bi se najčešće izveo DDOS (engl. *Distributed denial-of-service*) napad. Napadač mijenja izvorišnu adresu u IP zaglavlju te radi veliki broj zahtjeva na različite servise s odredišnom adresom

žrtve koja tada dobiva odgovor od svih servisa na koje su poslani zahtjevi kako bi zaštitio i blokirao valjan promet. Obrana od takve vrste napada se najčešće provodi na usmjerivačima analizom prometa kako bi se utvrdili nevažeći paketi.

Primjena protokola IPv4 je još uvijek u velikoj upotrebi usprkos tome što je broj dostupnih adresa davno iscrpljen. Daljnja upotreba protokola je omogućena zbog upotrebe NAT-a (engl. *Network Address Translation*). IPv6 nudi poboljšanja nad IPv4 mogućnošću jedinstvenog adresiranja više uređaja te podrškom da jedan uređaj istovremeno može pripadati više mreža upotrebom više IP adresa. Mogućnošću jedinstvenog adresiranja zbog većeg broja adresa jednostavnija je implementacija sustava s ravnopravnim sudionicima (engl. *Peer-to-peer network*).

4.5. Transportni sloj

Transportni sloj je odgovoran za komunikaciju između dva aplikacijska procesa preko mreže. Transportni protokoli definiraju vrata (engl. *port*) koja služe za identifikaciju aplikacijskih procesa na računalima. Najkorišteniji protokoli na transportnom sloju su UDP (engl. *User Data Protocol*) i TCP (engl. *Transmission Control Protocol*) od kojih svaki imaju određena vrata predefinirana za određene protokole aplikacijskog sloja. U nastavku su analizirani i uspoređeni protokoli TCP i UDP.

4.5.1. Analiza protokola

4.5.1.1. TCP

TCP je konečijsko orientirani protokol koji za uspostavu konekcije koristi trosmjerno rukovanje. Neka od glavnih svojstva TCP-a su pouzdani prijenos paketa, detekcija pogrešaka, ponovni prijenos kod isteka vremena te kontrola toka i zagruženja. Pouzdani prijenos paketa se postiže korištenjem sekvencijskih brojeva koji identificiraju svaki bajt podataka. Nakon slanja paketa pošiljatelj čeka na potvrdu od primatelja da je primio poslani paket. Detekcija pogrešaka je vezana uz potvrdu primatelja jer se tako može detektirati ako je neki od paketa izgubljen te se ponovno šalje. Pogreške se također detektiraju pomoću kontrolne sume u zaglavljima čime se utvrđuje ispravnost paketa. Ponovnim prijenosom kod isteka vremena se čeka određeno vrijeme na potvrdu primatelja o primljenom paketu te ako dođe do isteka vremena se paket ponovno šalje. Kontrolom toka i zagruženja se upravlja brzinom i veličinom slanja paketa kako bi se pouzdano mogli pošiljati paketi bez preopterećenja primatelja u slučaju male propusnosti mreže ili opterećenja primatelja.

4.5.1.2. UDP

UDP je za razliku od TCP-a beskonečnijski protokol koji ne sadrži svojstva TCP-a za pouzdani prijenos paketa osim detekcije pogrešaka korištenjem kontrolne sume. UDP sadrži četiri polja zaglavljia: izvorišna i odredišna vrata, duljina paketa i kontrolna suma. UDP je namijenjen brzim i vremenski osjetljivim namjenama.

4.5.2. Usporedba sigurnosnih mehanizama i primjena

Svojstva TCP-a omogućava pouzdani prijenos podataka između dvije krajnje točke. Tako se osigurava da informacije budu dostavljene bez pogrešaka i u cijelosti. Za razliku UDP nema mehanizme pouzdanog prijenosa podataka što rezultira brzom prijenosu jer ne postoji potreba za dodatnom obradom i kontrolom paketa.

Područja primjene UDP-a je u aplikacijama kojima je brzina prijenosa važnija od cjelovitosti i točnosti informacija poput video poziva, mrežnih igara i reprodukciji video snimaka. Područja primjene TCP-a je u aplikacijama gdje je pouzdanost informacija ključna poput bankovnih sustava, internet trgovini i elektroničkoj pošti.

4.6. Aplikacijski sloj

Aplikacijski sloj je zadnji sloj složaja Interneta stvari preko kojeg se izvodi izravna komunikacija između aplikacija. Protokoli ovog sloja omogućuju postavljanje komunikacijskih pravila između aplikacija kako bi način razmjene podataka bio standardiziran. Aplikacijski sloj sadrži jako veliki broj protokola čija namjena je s obzirom na primjenu jako različita. U nastavku su analizirana i uspoređena tri protokola aplikacijskog sloja koja su najviše zastupljena u Internetu stvari.

4.6.1. Analiza protokola

4.6.1.1. HTTP

HTTP (engl. *Hypertext Transfer Protocol*) je protokol na aplikacijskom sloju za distribuirane, kolaborativne, hipermedijske sustave te koristi TCP kao bazu za komunikaciju između poslužitelja i klijenta. HTTP je u svom začetku bio osmišljen kao protokol za razmjenu hiperteksta, dok je danas zaslužan za razmjenu raznih hipermedijskih sadržaja, tj. teksta, slike, zvuka i videa te kao takav ima najveći udio prometa na Internetu među aplikacijskim protokolima. HTTP koristi URI (engl. *Universal Resource Identifier*) shemu za identificiranje resursa na mrežnoj lokaciji te koristi vrata 80 TCP-a.

HTTP djeluje na temelju zahtjeva i odgovora gdje klijent zahtjeva resurs koristeći URI i definirane HTTP metode te na temelju zahtjeva dobiva odgovor od poslužitelja koji sadrži status odgovora, neobavezna polja zaglavlja te zahtijevani resurs u tijelu poruke ako status odgovora poprima format 2XX.

Zahtjevi osim tijela poruke također mogu sadržavati neobavezna polja zaglavlja koja daju više informacija vezana uz zahtjev poput zaglavlja koja imaju sigurnosne mehanizme: *Content-MD5* polje koje služi za provjeru integriteta poruke korištenjem algoritma MD5 za izračunavanje kontrolne sume, *Authorization* polje koje omogućuje autorizaciju korisnika na poslužitelju korištenjem korisničkog imena i lozinke kodirane *Base64* kodnom stranicom ili neke druge vrste autorizacije. HTTP ima nekoliko verzija protokola od kojih su trenutno najzastupljenije HTTP/1.1 i HTTP/2 koje u suštini ne nude prevelike razlike osim veće brzine verzije dva zbog boljeg upravljanja zahtjevima, kompresijom zaglavlja i bolje upotrebe TCP konekcija. Trenutno je u razvoju treća verzija protokola koja kao protokol transportnog sloja koristi UDP i kao zadano donosi šifriranje podataka.

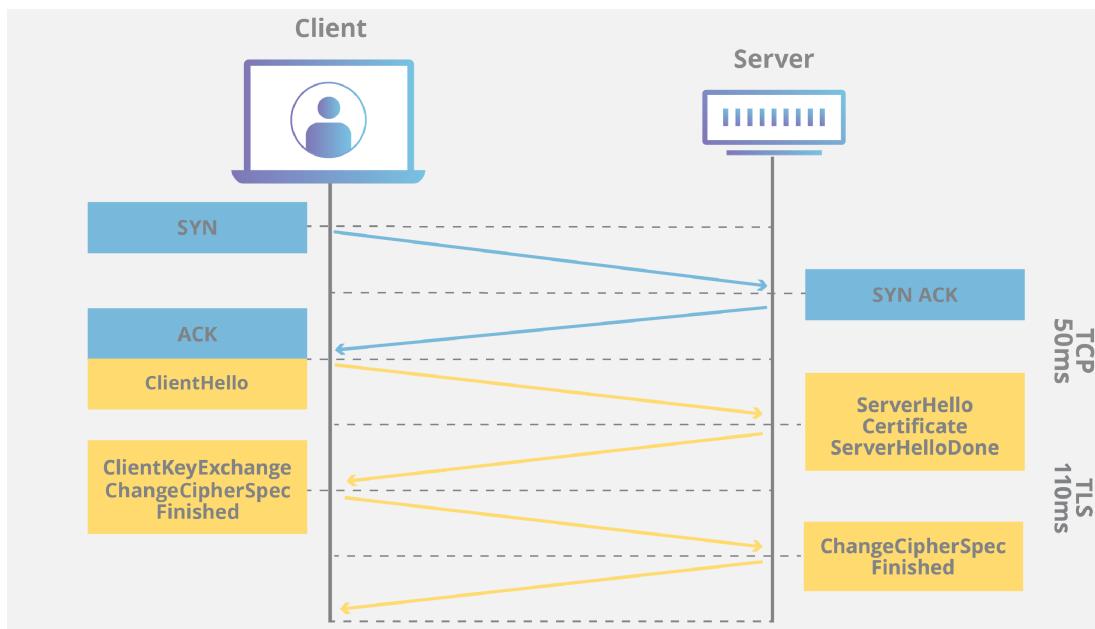
HTTP/S (engl. *Hypertext Transfer Protocol Secure*) je nadogradnja HTTP-a uz korištenje kriptografskog protokola TLS (engl. *Transport Layer Security*) za šifriranje i autentifikaciju. HTTPS kao i HTTP koristi URI shemu za identificiranje resursa na mrežnoj lokaciji, a umjesto vrata 80 koristi vrata 443 TCP-a. TLS podržava razne kriptografske algoritme za šifriranje te digitalne certifikate za autentifikaciju. Verzija TLS-a koja se preporučuje za korištenje je TLS 1.3 zbog pronađenih sigurnosnih propusta u prijašnjim verzijama. Postupak kojim se omogućava šifriranje i autentifikacija TLS-om se naziva rukovanje i dolazi nakon uspostava TCP konekcije te se izvodi u sljedećim koracima[28]:

1. **"Client hello" poruka:** Klijent inicira rukovanje slanjem "hello" poruke poslužitelju. Poruka sadržava verzije TLS-a i kriptografske algoritme koje klijent podržava te slučajno generirani niz znakova.
2. **"Server hello" poruka:** Poslužitelj odgovara klijentu s vlastitom "hello" porukom koja sadrži poslužiteljev digitalni certifikat, odabrani kriptografski algoritam i novi slučajno generirani niz znakova.
3. **Autentifikacija:** Klijent autentificira poslužitelja koristeći certifikacijsko tijelo koje je izdalo certifikat. Tako klijent potvrđuje da je poslužitelj onaj koji tvrdi da je i da klijent komunicira sa stvarnim vlasnikom domene.
4. **Tajna koja prethodi glavnoj:** Klijent ponovno šalje novi slučajno generirani

niz znakova, takozvana tajna koja prethodi glavnoj. Taj niz znakova je šifriran javnim ključem poslužitelja koji je bio dostavljen u digitalnom certifikatu.

5. **Korištenje privatnog ključa:** Poslužitelj dešifrira primljeni šifrirani niz znakova koristeći vlastiti privatni ključ.
6. **Kreiranje ključa sjednice:** Klijent i poslužitelj kreiraju simetrični ključ sjednice iz prvog klijentskog slučajnog niza znakova, poslužiteljevog slučajnog niza znakova i tajne koja prethodi glavnoj.
7. **Klijent je spreman:** Klijent šalje "finished" poruku koja je šifrirana s ključem sjednice.
8. **Poslužitelj je spreman:** Poslužitelj šalje "finished" poruku koja je šifrirana sa ključem sjednice.
9. **Postignuto je sigurno simetrično šifriranje:** Rukovanje je dovršeno i dalje se može izvoditi šifrirana komunikacija korištenjem simetričnog šifriranja.

Na sljedećoj slici je prikazan postupak TCP i TLS rukovanja potreban kako bi se uspostavila HTTP/S sjednica.



Slika 4.15: Prikaz TCP i TLS rukovanja[28]

4.6.1.2. CoAP

CoAP (engl. *Constrained Application Protocol*)^[3] je specijalizirani protokol za uređaje ograničenih resursa te kao protokol transportnog sloja koristi UDP na vratima 5683. Protokol je dizajniran za pouzdani prijenos podataka u okolinama male propusnosti i visoke razine zagušenja prometa upotrebom malih zaglavlja paketa i potrebi za maloj procesnoj snazi za obradu zahtjeva. CoAP kao i HTTP djeluje na temelju zahtjeva i dogovora između klijenta i poslužitelja, a koristi URI shemu za identificiranje resursa na mrežnoj lokaciji. Metode koje se koriste u zahtjevima su identične HTTP-u te se temelje na REST (engl. *Representational State Transfer*) arhitekturi. Metode zahtjeva su definirane kodnim brojem u zaglavlju poruke koje definira i kodni broj statusa odgovora. Kodni broj je veličine 8 bitova. Podržane metode su prikazane u tablici.

Tablica 4.3: Podržane metode CoAP zahtjeva

Metoda	Funkcionalnost	Svojstva
GET	dohvaća resurs na navedenom URI-u	sigurna, idempotentna
POST	stvara priloženi resurs na navedenom URI-u	nije sigurna ni idempotentna
PUT	ažurira ili stvara priloženi resurs na navedenom URI-u	nije sigurna, idempotentna
DELETE	briše resurs na navedenom URI-u	nije sigurna, idempotentna

CoAP podržava četiri vrste poruka:

CON(Confirmable Message): Svaka poruka treba dobiti odgovor ACK ili RESET kao potvrdu.

NON(Non-Confirmable Message): Poruke ne trebaju dobiti potvrdu primitka.

ACK(Acknowledgement Message): Potvrda da je specifična CON poruka zaprimljena.

RESET(Reset Message): Potvrda da je specifična CON ili NON poruka zaprimljena, ali je kontekst poruke nedovoljan da bi se poruka obradila.

CoAP nudi mogućnost promatranja resursa temeljen na preplati i objavi arhitekturi. Klijent registrira preplatu na određeni resurs korištenjem zaglavla opcija u kojem postavlja opciju *Observe:0*. Nakon toga se svakim ažuriranjem resursa šalje poruka na preplaćenog klijenta sve dok klijent ne otkaže preplatu slanjem opcije *Observe:1*. Na ovaj način se smanjuje broj potrebnih razmijenjenih poruka jer se izbjegava periodično slanje zahtjeva klijenta.

Kao što se HTTP osigurava upotrebom TLS-a preko TCP-a, tako CoAP koristi DTLS (engl. *Datagram Transport Layer Security*) za šifriranje i autentifikaciju. DTLS funkcioniра на истом principu као и TLS koji je objašnjen у HTTP odjeljku. CoAP с уključenim DTLS-om за razliku од HTTPS-a koristi UDP као protokol transportnog sloja на vratima 5684.

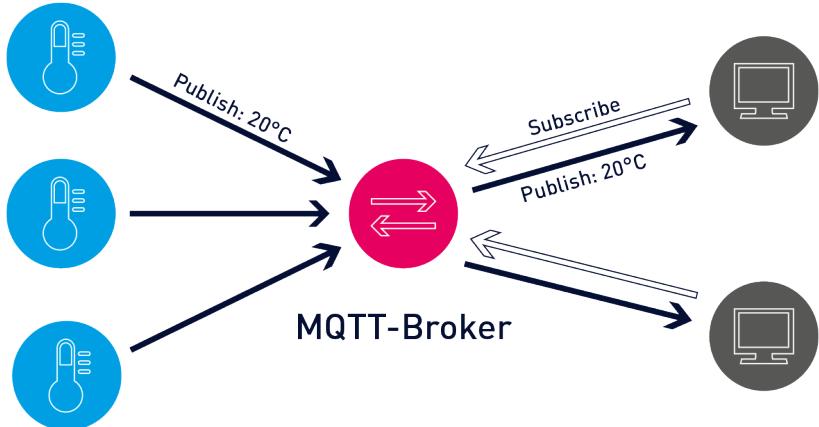
4.6.1.3. MQTT

MQTT (engl. *Message Queuing Telemetry Transport*) je lagani protokol temeljen на porukama, koristi TCP као protokol transportnog sloja на vratima 1883 te se bazira на topologiji PUSH/SUBSCRIBE. Arhitektura protokola MQTT sadrži dvije vrste sudionika: klijente i posrednike (engl. *broker*). Posrednik je poslužitelj s kojim klijenti komuniciraju putem poruka te posrednik proslijeđuje te poruke на ostale klijente. Tako klijenti ne komuniciraju s ostalim klijentima već putem posrednika. Svaki klijent može biti izdavač (engl. *publisher*), pretplatnik (engl. *subscriber*) ili oboje.

MQTT je protokol temeljen на događajima te tako nema periodičnog ili stalnog prijenosa podataka čime se smanjuje količina prijenosa. Izdavač šalje poruke jedino kada ima informacije за slanje, а posrednik proslijeđuje poruke pretplatnicima kada primi nove informacije. Još jedan način smanjenja količine prijenosa podataka je korištenjem strogo definiranom, malom konstrukcijom poruka. Svaka poruka sadrži zaglavje od samo dva bajta, ali može sadržavati i dodatno neobavezno zaglavje, dok je sadržaj poruke ograničen на 256MB.

Tri različite razine kvalitete usluge su dostupne kako bi se moglo odabrati između smanjenja količine i osiguravanja pouzdanosti prijenosa podataka. Nulta razina definira slanje poruka pretplatnicima bez potvrde je li poruka primljena, kod prve razine posrednik čeka potvrdu poruke te ako dođe до isteka vremena poruka se ponovno šalje te tako pretplatnik može poruku primiti više puta. Druga razina koristi četverostruko rukovanje између klijenta i posrednika kako bi se osiguralo da je poruka primljena и то točno jedan put. Za prvu i drugu razinu poruke se spremaju за klijente koji nisu trenutno dostupni te se ponovno šalju kada klijent postane ponovno dostupan. На slje-

dećoj slici je prikazana topologija sudionika protokola MQTT.



Slika 4.16: Topologija sudionika protokola MQTT[22]

Poruke koje izdavači objave se temelje na temama. Teme su strukturirane u hijerarhiju koja teme odvaja znakom "/" kao i kod datotečnih sustava. Primjer takve strukture je "**FER/zgrada-C/kat-7/soba-04/senzor/temperatura**" koja dozvoljava pretplatniku da navede kako želi primati samo informacije o temperaturi sobe četiri sedmog kata FER-ove C zgrade ili da se želi pretplatiti na sve senzore na FER-u koristeći sintaksu "FER/+/-/+/-/senzor/+". Kod objave prve poruke na temu koja ne postoji ta tema se automatski definira na posredniku bez prethodne konfiguracije.

MQTT podržava 14 vrsta poruka od kojih su osam potvrde, a šest zahtjevi opisani u nastavku.

PUBLISH: izdavač šalje podatke teme na posrednika. Ako tema ne postoji, ona se stvara na posredniku. Ovisno o definiranoj razini kvalitete usluge posrednik šalje potvrdu o primitku poruke.

SUBSCRIBE/UNSUBSCRIBE: pretvara klijenta u pretplatnika na temu ili uklanja pretplatu na temu. Pretplate mogu biti vezane uz specifičnu temu, sve teme određene razine hijerarhije ili samo određene dijelove određenih razina hijerarhije tema. Kao odgovor klijent dobiva potvrdu od posrednika.

PING: Klijent provjerava je li TCP konekcija prema posredniku još uvijek živa. Kao odgovor se dobiva potvrda ako je posrednik dostupan i TCP konekcija živa.

CONNECT/DISCONNECT: Klijent šalje poruku posredniku za otvaranje ili zatvaranje konekcije. Kao odgovor se dobiva potvrda od posrednika o uspostavi konekcije, a kod zatvaranja konekcije se ne šalje potvrda.

Originalan cilj protokola MQTT je bilo smanjenje količine i efikasnost prijenosa podataka kako bi se mogla omogućiti komunikacija putem skupih i nepouzdanih komunikacijskih kanala poput satelitskog prijenosa. Tako kod razvoja protokola sigurnost nije bila primarna briga. Unatoč tome MQTT nudi nekoliko sigurnosnih mehanizama poput korištenje korisničkog imena i lozinke za autorizaciju klijenta kod posrednika. Nedostatak kod implementacije autorizacije je što se korisničko ime i lozinka prenose u nešifriranom tekstualnom formatu. Problem šifriranja i autentifikacije je riješeno korištenjem protokola TLS te se tada protokolu pristupa putem TCP vrata 8883.

4.6.2. Usporedba sigurnosnih mehanizama i primjena

Sva tri navedena protokola nemaju vlastite mehanizme koji bi omogućavali šifriranje podataka i autentifikaciju te za to koriste protokol TLS kao nadogradnju. HTTP i MQTT koriste TLS s TCP-om kao protokol transportnog sloja dok CoAP koristi DTLS koji koristi UDP za protokol transportnog sloja. TLS i DTLS su funkcijски identični protokoli s jedinom razlikom u korištenom transportnog protokolu. HTTP i MQTT također imaju neobaveznu podršku za autorizacijom klijenta korištenjem korisničkog imena i lozinke.

HTTP je u svojoj primjeni puno fleksibilniji zbog mogućnosti prijenosa bilo koje vrste podataka koji uključuju hipertekstualne datoteke, slike, video i audio zapisa te kao takav je najzastupljeniji protokol aplikacijskog sloja na Internetu. Za razliku protokoli MQTT i CoAP su dizajnirani kao lagani protokoli za uređaje i mreže ograničenih resursa. Tako je glavna namjena tih protokola kod praćenja očitanja i jednostavnih događaja pretežito prisutnih u Internetu stvari.

5. Sustav za praćenje tjelesne temperature

Kako bi se olakšalo praćenje kretanja tjelesne temperature te smanjila potreba za ručnim mjerjenjem korištenjem termometara, razvijen je sustav za praćenje tjelesne temperature. Cilj ovog sustava je omogućiti korištenje prijenosnog termometra koji se nalazi prilipljen uz tijelo korisnika kako bi se mogao pratiti trend tjelesne temperature u razmaku minutnog intervala. Pregled podataka i upravljanje uređajima se odvija putem internet preglednika kako bi se s udaljenog mjesta mogao imati nadzor nad temperaturom korisnika.

5.1. Korišteni razvojni alati i uređaji

5.1.1. Golang

Golang[18] je proceduralno i statičko pisani programski jezik sa sintaksom sličnom programskom jeziku C. Nudi bogati izbor standardnih biblioteka, dinamičko upravljanje memorijom i mogućnosti za visoku razinu konkurentnosti. Jezik je razvijen 2007. godine u Googleu te je 2009. objavljen kao jezik otvorenog koda, a zadnja verzija je 1.16.5 te je jedan od programskih jezika koji je u zadnjih nekoliko godina dosegao visoku razinu popularnosti. Golang se najaviše koristi kao HTTP poslužitelj i jedan je od jezika koji je popularizirao razvoj aplikacija temeljenih na mikroservisnoj arhitekturi.

5.1.2. Fiber

Fiber[16] je Golang razvojni okvir namijenjen za HTTP poslužitelje. Dizajniran je za jednostavan i brz razvoj HTTP poslužitelja s naglaskom na performansama i velikom broju konkurentnih korisnika.

5.1.3. React

React[26] je JavaScript biblioteka za razvoj korisničkih sučelja. Omogućuje jednostavan razvoj interaktivnih sučelja temeljenih na komponentama koje upravljaju vlastitim stanjima i na temelju promjena stanja reagiraju na promjene kako bi se ažurirali ili prikazali potrebni dijelovi sučelja.

5.1.4. PostgreSQL

PostgreSQL[24] je objektno-relacijska baza podataka otvorenog koda. PostgreSQL je u razvoju više od 30 godina te ima reputaciju kao pouzdana, robusna baza podataka visokih performansi s velikim brojem značajki.

5.1.5. Docker

Docker[14] je platforma za kreiranje i pokretanje aplikacija na ponovljiv način korištenjem kontejnera. Kontejneri dozvoljavaju pakiranje aplikacija sa svim bibliotekama, ovisnostima i programskim kodom potrebnim da se aplikacija izvršava kao jedan paket, tj. slika. Kontejneri su lagani virtualni operativni sustavi koji umjesto korištenja cijelog operativnog sustava kao jedan paket iskorištavaju jezgru operativnog sustava domaćina te tako smanjuju veličinu slike potrebne za pokretanje.

5.1.6. Nginx

Nginx[23] je HTTP i zastupnički (engl. *reverse proxy*) poslužitelj, raspodjelitelj opterećenja i priručno spremište (engl. *cache*). Nginx je programska podrška otvorenog koda te je jedan o najpopularnijih *web* poslužitelja na Internetu gdje se najčešće koristi kao raspodjelitelj opterećenja.

5.1.7. ESP32

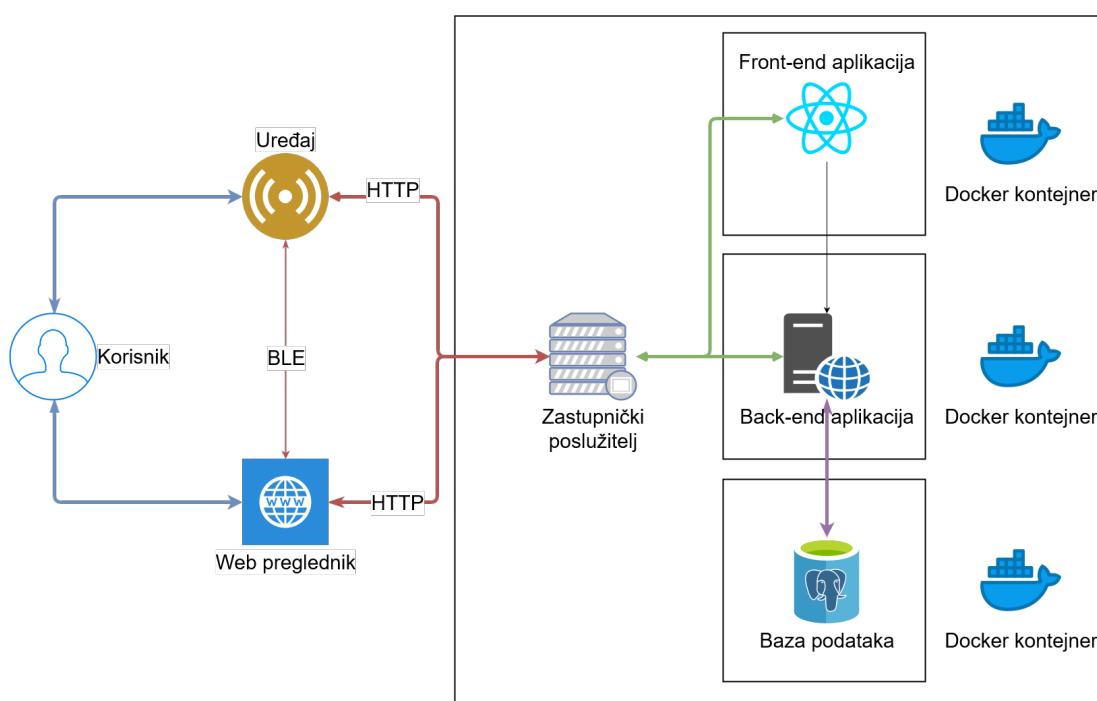
Joy-IT NodeMCU-ESP32[15] je varijanta ESP32 integriranog računala s mikrokontrolerom od proizvođača Joy-IT. ESP32 nudi dva komunikacijska modula za Wi-Fi 802.11 b/g/n i Bluetooth s podrškom za BLE. Priključni pinovi dozvoljavaju laku integraciju analognih senzora, a micro USB priključak omogućuje lako priključivanje vanjskih prijenosnih baterija za napajanje uređaja. ESP32 je energetski učinkovito računalo s podrškom za duboki san kako bi se uvelike smanjila potrošnja energije kada uređaj ne treba vršiti nikakvu funkciju.

5.2. Arhitektura i dizajn sustava

Sustav za praćenje tjelesne temperature se sastoji od 4 glavne komponente:

- baze podataka,
- back-end aplikacije,
- front-end aplikacije,
- uređaja.

Baza podataka, back-end i front-end aplikacije se izvršavaju u zasebnim docker kontejnerima kako bi se olakšalo postavljanje u produkciju. Docker kontejneri dozvoljavaju da se sve komponente izvršavaju u jednakoj okolini neovisno o računalu na kojem se izvršavaju te se tako izbjegavaju nekompatibilnosti koje pojavljuju kada se program izvršava u različitim operacijskim sustavima. Pristup front-end i back-end aplikacijama se izvodi preko Nginx-a kao zastupničkog poslužitelja koji preusmjerava zahtjeve na određenu aplikaciju ovisno o URL-u te omogućuje uspostavljanje sigurne HTTPS veze između klijenta i aplikacija. Arhitektura sustava je prikazana na sljedećoj slici.

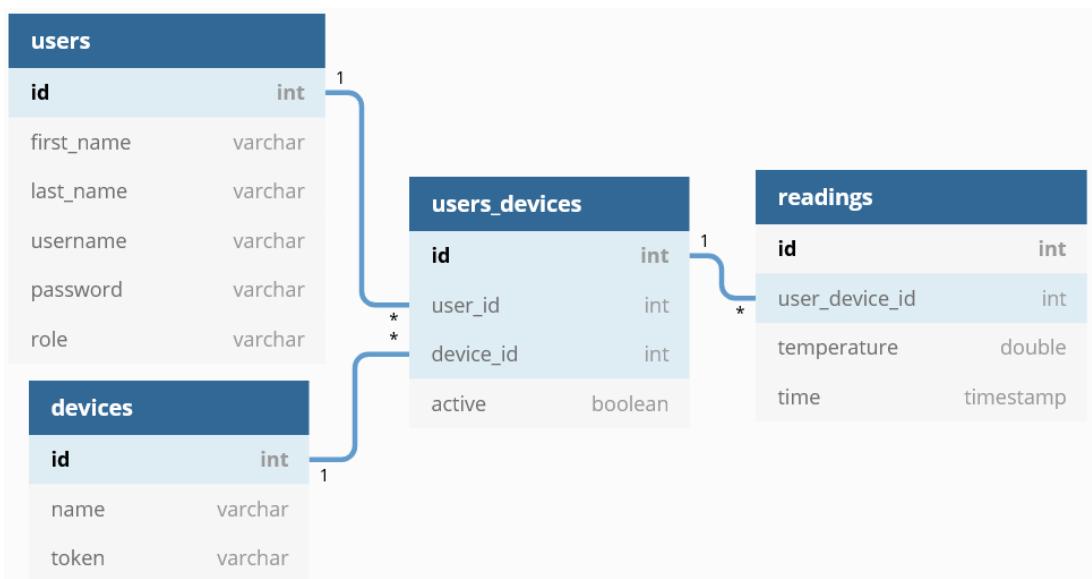


Slika 5.1: Arhitektura sustava

Pristup svim funkcionalnostima sustava se izvodi preko internet preglednika putem kojeg se radi pregled informacija i upravljanje korisnicima te konfiguracija i upravljanje uređajima.

5.2.1. Baza podataka

Podaci sustava se spremaju u PostgreSQL bazu podataka koja se izvršava u Docker kontejneru. Baza se sastoji od 4 tablice te je relacijska shema baze prikazana na sljedećoj slici.



Slika 5.2: Baza podataka

Tablica `users` predstavlja korisnike sustava te ovisno o ulozi (engl. `role`) definira administratore i korisnike za koje se promatra temperatura. Tablica `devices` predstavlja uređaje koji se koriste za opažanje temperature te je definirano ime i pristupni `token` uređaja. Tablica `users_devices` predstavlja vezu između korisnika i uređaja kod koje stupac `active` određuje za koji uređaj je trenutno pridijeljen korisnik. Jedan uređaj ima pridijeljenog jednog korisnika te je tada stupac `active` istinit. Tablica `readings` predstavlja očitanja koja uređaj opaža te su definirani stupci koji predstavljaju temperaturu i vrijeme kada je temperatura očitana. Definicija tablica i relacijskih veza se nalazi u datoteci: `implementation/database/db.sql`

5.2.2. Back-end aplikacija

Back-end aplikacija je pisana u Golang programskom jeziku te se koristi Fiber razvojni okvir za razvoj HTTP poslužitelja. Aplikacija se izvršava u Docker kontejneru te se zahtjevi na poslužitelj preusmjeravaju s Nginx-a. Zahtjevi na aplikaciju dolaze s dva izvora: front-end aplikacije koja se izvršava u internet pregledniku i uređaja. Aplikacija je dizajnirana po REST arhitekturi gdje se pristup određenim resur-

simu odvija na temelju URL-a definiranog u zahtjevu. Aplikacija obrađuje zahtjeve te komunicira s bazom podataka kako bi dohvatala ili spremila potrebne podatke. Programska podrška aplikacije se sastoji od sljedećih datoteka koje se nalaze na putanji: `implementation/backend/`

go.mod: definira verziju Golanga i sve vanjske biblioteke potrebne za izvršavanje i prevođenje programske potpore.

go.sum: definira URL, verzije i identifikacijske oznake biblioteka koje su potrebne za izvršavanje i prevođenje programske potpore.

main.go: predstavlja prvobitni dio aplikacije u kojem se inicijaliziraju svi potrebni dijelovi poput konekcija s bazom podataka, HTTP poslužitelj, rute koje obrađuju određene funkcije na temelju REST arhitekture i posrednička funkcija za autorizaciju korisnika.

routes.go: definira rute i funkcije koje obrađuju zahtjeve na određenom URL-u.

auth.go: Definira funkcije za logiranje i registraciju korisnika te posredničku funkciju za autentifikaciju i autorizaciju korisnika kada pristupa određenim resursima sustava.

handlers.go: definira sve funkcije REST arhitekture koje obrađuju zahtjeve za određenim resursima sustava.

models.go: definira strukture tablica baze podataka i svih zahtjeva i odgovora.

5.2.3. Front-end aplikacija

Front-end aplikacija je pisana u JavaScript razvojnog okviru React. Sama aplikacija se izvršava u internet pregledniku dok se u zasebnom docker kontejneru na poslužitelju odvija dohvati potrebnih datoteka. Datoteke aplikacije se nalaze na putanji: `implementation/frontend/`. U `package.json` datoteci se nalaze definirane biblioteke koje aplikacija koristi, podržani internet poslužitelji i skripte za izvršavanje. U `src` mapi se nalazi sva potrebna programska podrška koja se izvršava u internet pregledniku. Neke od datoteka i mapa u `src` folderu su opisane u nastavku.

index.js: pristupna datoteka aplikacije koja enkapsulira ostatak aplikacije.

index.css: definira CSS stilove za osnovne HTML klase.

App.js: glavni dio aplikacije koji definira usmjeritelj na ostale komponente ovisno o URL-u i o tome je li korisnik prijavljen.

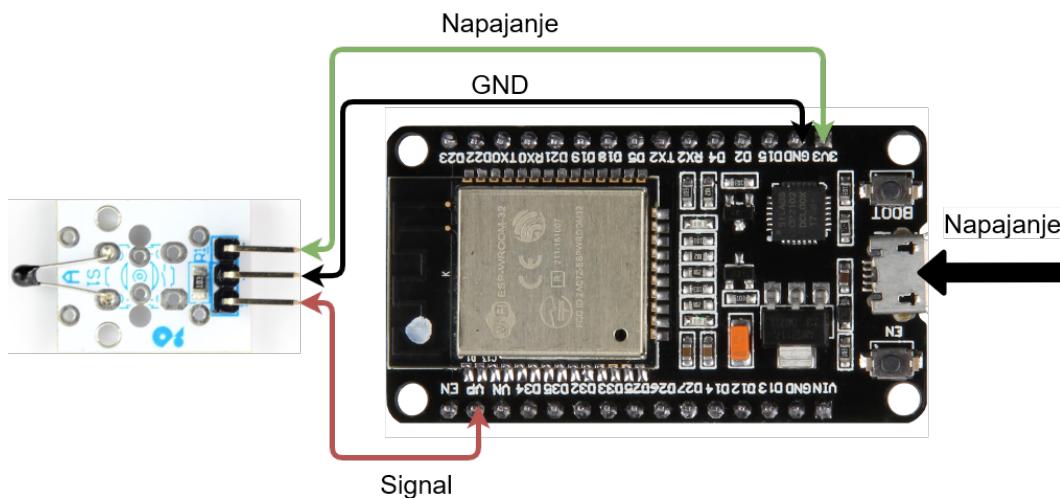
App.css: definira CSS stilove za HTML klase komponenata.

components/*: ova mapa definira komponente koje se koriste u aplikaciji poput komponenti za prijavu, registraciju, navigacijsku traku, konfiguraciju uređaja i prikaz grafa i tablice očitane temperature.

Aplikacija komunicira s back-end aplikacijom putem protokola HTTP kako bi dohvila potrebne podatke te je potrebno prvobitno biti prijavljen kako bi se dohvaćanje podataka moglo odvijati. Za potrebe konfiguracije uređaja koristi se komunikacija protokolom BLE. Konfiguracija uređaja se može odvijati samo putem internet preglednika koji podržavaju protokol BLE i koji se izvršavaju na računalu s kompatibilnim Bluetooth komunikacijskim modulom.

5.2.4. Uredaj za mjerjenje temperature

Za potrebe mjerjenja temperature se koristi Joy-IT NodeMCU-ESP32 integrirano računalo s Velleman VMA320 analognim senzorom za toplinu. Napajanje uređaja se vrši putem vanjske prijenosne baterije koja je priključena na micro USB priključak. Analogni senzor je povezan s uređajem putem 3 priključna pina za ulaz, napajanje i uzemljenje. Shema spajanja uređaja sa senzorom i baterijom je prikazana na sljedećoj slici.



Slika 5.3: Shema spajanja

Uredaj komunicira putem dva komunikacijska protokola. Putem protokola Wi-Fi komunicira s internet poslužiteljem na kojem se izvršava back-end aplikacija kako bi

poslao očitanja temperature. Putem protokola BLE komunicira s internet preglednikom na kojem se izvršava front-end aplikacija za potrebe konfiguracije uređaja. Kako bi uređaj mogao komunicirati s back-end aplikacijom, potrebno je konfigurirati parametre opisane u nastavku.

ID: identifikator uređaja.

Token: pristupni token kojim se autentificira uređaj.

Host: domena internet poslužitelja na kojem se izvršava back-end aplikacija.

SSID: identifikator pristupne Wi-Fi točke.

Password: lozinka za pristupnu Wi-Fi točku.

Programska podrška uređaja se nalazi u datoteci: `implementation/esp32/module.ino`

5.3. Opis rada sustava

Sustav za praćenje tjelesne temperature sadrži sedam glavnih obrazaca uporabe koji su opisani u nastavku.

5.3.1. Registracija korisnika

Kako bi korisnik dobio pristup sustavu potrebno je da se prvo registrira. Registraciji je moguće pristupiti ako korisnik nije već prijavljen na sustav. Registracija se provodi upisivanjem željenog korisničkog imena, imena i prezimena korisnika te željene lozinke. Kada korisnik ispuni podatke obrasca i pritisne gumb *Registriraj* podaci se šalju na back-end aplikaciju gdje se spremaju u bazu podataka.

5.3.2. Prijava korisnika

Nakon registracije korisnika potrebno je da se korisnik prijavi kako bi mogao pristupiti sustavu. Prijava se vrši ispunjavanjem obrasca u koji je potrebno unijeti korisničko ime i lozinku. Nakon pritiska na gumb *Prijavi* podaci obrasca se šalju na back-end aplikaciju gdje se odrađuje autorizacija korisnika na temelju podataka iz baze. Ako su podaci točni, korisnik se uspješno prijavio na sustav te ga se preusmjerava na dio sustava koji je dostupan samo prijavljenim korisnicima.

5.3.3. Kreiranje korisnika

Kreiranje korisnika se odvija unošenjem podataka o imenu i prezimenu korisnika. Tako se kreira korisnik uloge *Pacijent* koji nema pristup sustavu nego mu se samo može dodjeliti uređaj i pratiti podaci o njegovoj temperaturi.

5.3.4. Uređivanje korisnika

Uređivanje korisnika se odvija promjenom podataka i dodjeljivanjem uređaja koji će od toga trenutka biti povezan s tim korisnikom.

5.3.5. Konfiguracija uređaja

Konfiguracija uređaja se odvija prvo povezivanjem putem protokola BLE s uređajem. Kako bi se mogao konfigurirati uređaj, potrebno je unijeti pristupnu lozinku za uređaj. U obrazac se upisuju pristupni token, domena poslužitelja, SSID i lozinka Wi-Fi pristupne točke koje se zatim šalju na uređaj s identifikatorom uređaja. Osim na uređaj, ti podaci se šalju i na back-end aplikaciju. Konfiguracijom uređaja počinje očitavanje tjelesne temperature.

5.3.6. Očitavanje tjelesne temperature

Uređaj svake minute očitava temperaturu sa senzora te šalje zahtjev s podacima o identifikatoru uređaja, očitanoj temperaturi i pristupnom tokenu na back-end aplikaciju. Aplikacija provjerava token i identifikator u bazi podataka i, ako podaci zahtjeva odgovaraju podacima u bazi, aplikacija sprema podatke o izmjerenoj temperaturi i vremenu mjerena u bazu.

5.3.7. Pregled trenda kretanja temperature

Prijavljeni korisnik može pregledavati trend kretanja temperature za korisnike koji postoje u sustavu. Kod pregleda se prikazuje linijski graf gdje x-os predstavlja vrijeme, a y-os temperaturu. Uz graf se prikazuje i tablica s vremenom i temperaturnim očitajima.

5.4. Sigurnosna analiza sustava

Analiza sustava će se provoditi kroz zahtjeve vezane uz sigurnost i privatnost koji su definirani u odjeljku 3.1.

5.4.1. Tajnost

Tajnost podataka je očuvana korištenjem autentifikacije i autorizacije korisnika na sustavu. To se provodi prijavom korisnika na sustav kako bi mogao imati uvid u podatke o ostalim korisnicima i temperaturnim očitanjima. Kako bi se očuvala tajnost podataka u prijenosu, koristi se protokol HTTP/S za komunikaciju između internet preglednika i poslužitelja te uređaja i poslužitelja. Komunikacija između internet preglednika i uređaja je također šifrirana korištenjem BLE sigurnosnih mehanizama, a postupak uparanja se provodi *Passkey* metodom kod koje je potrebno unijeti pristupnu lozinku koja je definirana na uređaju.

5.4.2. Cjelovitost

Cjelovitost informacija se omogućava korištenjem protokola HTTP/S koji uporabom transportnog protokola TCP omogućuje komunikaciju bez gubitka informacija, a korištenjem kriptografskog protokola TLS se osigurava točnost podataka.

5.4.3. Raspoloživost

Raspoloživost cijelog sustava ovisi o dostupnosti poslužitelja na kojem se izvršava cijeli sustav. Ako dođe do prekida rada poslužitelja tada nije moguće pristupiti podacima i spremati podatke o očitanjima. Veći stupanj raspoloživosti se može ostvariti korištenjem dodatne redundancije poslužitelja.

5.4.4. Vjerodostojnjost

Vjerodostojnjost poslužitelja se postiže korištenjem certifikata pomoću kojeg se autentificira poslužitelj. Vjerodostojnjost uređaja se utvrđuje pomoću pristupnog tokena koji se nalazi u zahtjevu, a dodijeljen mu je od sustava. Vjerodostojnjost korisnika se utvrđuje korištenjem pristupnog korisničkog imena i lozinke.

5.4.5. Neporicanje

Neporicanje se postiže bilježenjem izvršenih aktivnosti u sustavu. Bilježi se vrijeme, korisnik i podaci kojima je korisnik pristupao. Tako se naknadno mogu utvrditi aktivnosti korisnika.

5.4.6. Odgovornost

Odgovornost u sustavu se postiže istim mehanizmom bilježenja kao i neporicanje. Tako se postiže odgovornost korisnika o provođenim aktivnostima.

5.4.7. Pouzdanost

Konzistentno ponašanje sustava se postiže uporabom REST arhitekture za dohvata podataka koja omogućava korištenje jednostavnih funkcija za obradu svih zahtjeva. Pouzdanost kod uređaja se postiže jednostavnom logikom koja obavlja očitavanja, povezivanje i prijenos informacija na poslužitelj.

6. Zaključak

Razvoj pristupnih mreža i energetski učinkovitih uređaja malih dimenzija omogućili su pojavu Interneta stvari u našoj svakodnevici. Pametni uređaji su sve više dio naših života te nam nude uvid u informacije i automatizaciju različitih procesa s kojima se svakodnevno susrećemo.

Dostupnost informacija o razvojnim alatima i niska cijena uređaja omogućuju jednostavan i brz razvoj pametnih uređaja i sustava Interneta stvari čime se proširuje broj različitih primjena. Tako se međuostalom dobiva konkurentnost na tržištu i veća integracija pametnih uređaja u živote korisnika.

Razvoj Interneta stvari donosi i mnoge izazove koji su vezani uz prirodu takvih sustava među kojima su raspodijeljenost, heterogenost, integracija, sigurnost i privatnost. Svi ti izazovi su međusobno povezani te je kod planiranja i razvoja sustava potrebno обратiti pažnju na njih, što se nekada očituje u zanemarivanju nekih od tih izazova.

Sigurnost i privatnost korisnika su izazovi s kojima se proizvođači najčešće susreću. Brz razvoj rješenja Internet stvari kako bi se proizvod prvi stavio na tržište uzrokuje razne sigurnosne propuste u sustavima čime se izravno ugrožava sigurnost i privatnost korisnika. Nedovoljna informiranost razvijatelja o sigurnosnim praksama uz brz razvoj također utječe na krajnju sigurnost sustava.

Sigurnosni propusti ne utječu samo na privatnost korisnika već i na sigurnost šire zajednice. Napadi na sustave Interneta stvari omogućuju onesposobljavanje raznih industrijskih postrojenja čiji sigurnosni mehanizmi sprječavaju nastanak ekoloških katastrofa.

Kako bi se osigurala sigurnost sustava i privatnost podataka, potrebna je dostupnost informacija vezanih uz uređaje i servise Interneta stvari. Dostupnost informacija poput otvorenih i dostupnih implementacijskih detalja dozvoljava široj razvojnoj zajednici istraživanje i konačno obznanjenje otkrivenih propusta u sustavima. Informiranjem javnosti omogućuje se korisnicima da donešu informiranu odluku o proizvodima koje koriste i tako zaštite sebe i svoju okolinu.

LITERATURA

- [1] Wi-Fi Alliance. URL <https://www.wi-fi.org>.
- [2] The "Only" Coke Machine on the Internet, Jun 1998. URL https://www.cs.cmu.edu/~coke/history_long.txt.
- [3] Constrained Application Protocol (CoAP), Mar 2012. URL <https://tools.ietf.org/id/draft-ietf-core-coap-03.html#opt>.
- [4] Overview of the Internet of things ITU-T Y.4000/Y.2060 (06/2012), Jun 2012. URL <http://handle.itu.int/11.1002/1000/11559>.
- [5] Cisco (2014) The Internet of Things Reference Model. - References - Scientific Research Publishing, Jun 2014. URL http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf.
- [6] UREDBA (EU) 2016/679 EUROPSKOG PARLAMENTA I VIJEĆA, 2016. URL <https://eur-lex.europa.eu/legal-content/HR/TXT/HTML/?uri=CELEX:32016R0679&from=HR>.
- [7] Number of IoT devices 2015-2025 | Statista, Nov 2016. URL <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide>.
- [8] Owasp top ten 2017, May 2017. URL <https://owasp.org/www-project-top-ten/2017>.
- [9] The Internet of Things with ESP32, Feb 2018. URL <http://esp32.net>.
- [10] Owasp internet of things, 2018. URL <https://owasp.org/www-project-internet-of-things/>.

- [11] Introducing the Raspberry Pi 4 - Raspberry Pi Spy, Jun 2019.
URL <https://www.raspberrypi-spy.co.uk/2019/06/introducing-the-raspberry-pi-4>.
- [12] Arduino Uno Rev3 | Arduino Official Store, Jun 2021. URL <https://store.arduino.cc/arduino-uno-rev3>.
- [13] Bluetooth® Technology Website, Jun 2021. URL <https://www.bluetooth.com>.
- [14] Empowering App Development for Developers | Docker, Jun 2021. URL <https://www.docker.com>.
- [15] NodeMCU ESP32 - Joy-IT, Jun 2021. URL <https://joy-it.net/en/products/SBC-NodeMCU-ESP32>.
- [16] Fiber, Jun 2021. URL <https://gofiber.io>.
- [17] Fipy - Pycom, Jun 2021. URL <https://pycom.io/product/fipy>.
- [18] The Go Programming Language, Jun 2021. URL <https://golang.google.cn>.
- [19] End Device Activation, May 2021. URL <https://www.thethingsnetwork.org/docs lorawan/end-device-activation>.
- [20] Homepage - LoRa Alliance®, Apr 2021. URL <https://lora-alliance.org>. [Online; accessed 27. Jun. 2021].
- [21] LoRaWAN Architecture, Jun 2021. URL <https://www.thethingsnetwork.org/docs lorawan/architecture>.
- [22] What is MQTT? Definition and Details, Jun 2021. URL <https://www.paessler.com/it-explained/mqtt>.
- [23] nginx, Jun 2021. URL <https://nginx.org/en>.
- [24] PostgreSQL, Jun 2021. URL <https://www.postgresql.org>.
- [25] Raspberry Pi Pico (Latest & original) buy online at Low Price in India - ElectronicsComp.com, Jun 2021. URL <https://www.electronicscomp.com>.

com/development-board-programmer/raspberry-pi/
raspberry-pi-boards/raspberry-pi-pico-india.

- [26] React, Jun 2021. URL <https://reactjs.org>.
- [27] Sensor board for Micro:bit, Jun 2021. URL
<https://shop.mchobby.be/en/microbit/1593-sensor-board-for-microbit-3232100015937-kitronik.html>.
- [28] What happens in a TLS handshake? | SSL handshake, Jun 2021. URL <https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake>.
- [29] Waspmove » The sensor platform to develop IoT projects - Libelium, Jun 2021. URL <https://www.libelium.com/iot-products/waspmove>.
- [30] ZigBee module FZB5200, Jun 2021. URL <https://www.aliexpress.com/item/32812739871.html>.
- [31] Noah Apthorpe, Dillon Reisman, i Nick Feamster. A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic. *arXiv*, May 2017. URL <https://arxiv.org/abs/1705.06805v1>.
- [32] Sheila Frankel, Bernard Eydt, Les Owens, i Karen Scarfone. Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i. Feb 2007. URL <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-97.pdf>.
- [33] Tim Greene. Study: Most enterprise IoT transactions are unencrypted. *Network World*, May 2019. URL <https://www.networkworld.com/article/3396647/study-most-enterprise-iot-transactions-are-unencrypted.html>.
- [34] Jaysson Hollingshead. The 7 Biggest Data Breaches of All Time | InfoSec Insights, Sep 2019. URL <https://sectigostore.com/blog/the-7-biggest-data-breaches-of-all-time>.
- [35] Pavle Skočir Ivana Podnar Žarko, Mario Kušek. Internet stvari, 2. predavanje. Zavod za telekomunikacije, Fakultet elektrotehnike i računarstva, 2020.

- [36] Daryna Kacherovska. How Secure Is the BLE Communication Standard? *Dzone*, Aug 2019. URL <https://dzone.com/articles/how-secure-is-the-ble-communication-standard>.
- [37] Bernard Meyer. After analyzing 15 billion passwords, these are the most common phrases people use | CyberNews, May 2021. URL <https://cybernews.com/best-password-managers/most-common-passwords>.
- [38] Mark Patel, Jason Shangkuan, i Christopher Thomas. What's new with the Internet of Things? *McKinsey & Company*, Jan 2018. URL <https://www.mckinsey.com/industries/semiconductors/our-insights/whats-new-with-the-internet-of-things>.
- [39] The Raspberry Pi Foundation. Buy a Raspberry Pi 4 Model B – Raspberry Pi, Jun 2021. URL <https://www.raspberrypi.org/products/raspberry-pi-4-model-b>.
- [40] The Raspberry Pi Foundation. Buy a Raspberry Pi Pico – Raspberry Pi, Jun 2021. URL <https://www.raspberrypi.org/products/raspberry-pi-pico>.

POPIS SLIKA

2.1. Nova dimenzija komunikacije predstavljena u Internetu stvari[4]	2
2.2. Ciscov referentni model Interneta stvari[5]	3
2.3. Broj povezanih uređaja Interneta stvari[7]	11
2.4. Potencijalni ekonomski utjecaj Interneta stvari kroz različita područja primjene za 2025. godinu. izražen u bilijunima američkih dolara[38] .	11
4.1. Protokolni složaj Interneta stvari[35]	23
4.2. Moduli uređaja Interneta stvari[35]	24
4.3. Senzorska pločica s mogućnošću očitanja razine buke, temperature i luminacijskog intenziteta[27]	25
4.4. ZigBee komunikacijski modul[30]	26
4.5. Raspberry Pi 4 Model B[11]	27
4.6. Raspberry Pi Pico[25]	27
4.7. Arduino Uno Rev3[12]	28
4.8. Libelium WaspMote[29]	29
4.9. Joy-IT NodeMCU-ESP32[15]	29
4.10. Pycom FiPy[17]	30
4.11. Faze uspostava i raskida veze korištenjem PSK metode[32]	34
4.12. Uparivanje uređaja protokolom BLE [36]	36
4.13. Topologija LoRaWAN arhitekture[21]	38
4.14. Tijek LoRaWAN 1.1 OTAA postupka[19]	41
4.15. Prikaz TCP i TLS rukovanja[28]	48
4.16. Topologija sudionika protokola MQTT[22]	51
5.1. Arhitektura sustava	55
5.2. Baza podataka	56
5.3. Shema spajanja	58

POPIS TABLICA

3.1. Primjer DNS upita napravljenih od strane uređaja [31]	18
4.1. Varijante protokola Wi-Fi	32
4.2. Usporedba sigurnosnih Wi-Fi standarda	33
4.3. Podržane metode CoAP zahtjeva	49

Analiza i usporedba sigurnosnih mehanizama u Internetu stvari

Sažetak

Internet stvari je sve rašireniji u svakodnevici ljudi u kojoj pametni uređaji dozvoljavaju automatizaciju i kreiranje novih procesa. Stoga je sustave Interneta stvari potrebno osigurati dostupnim sigurnosnim mehanizmima. Kroz rad se obrađuju definicija, referentni model koji opisuje slojeve sustava, područja primjene i trendovi Interneta stvari. Definiraju se zahtjevi vezani uz sigurnost i privatnost sustava i korisnika te najčešći sigurnosni propusti u takvim sustavima. Provodi se analiza i komparacija sigurnosnih mehanizama određenih uređaja i protokola složaja Interneta stvari. Opisuje se implementacija sustava za praćenje kretanja tjelesne temperature, njegova arhitektura, dijelovi sustava te se provodi sigurnosna analiza cijelog sustava.

Ključne riječi: Internet stvari, Informacijska sigurnost, ESP32, Wi-Fi, Bluetooth Low Energy, LoRaWAN, IP, TCP, UDP, HTTP, TLS, CoAP, MQTT, Analiza, Usporedba

Analysis and comparison of IoT security mechanisms

Abstract

The Internet of Things is becoming more widespread in people's everyday lives, where smart devices allow automation and creation of new processes. Therefore, IoT systems need to be secured with available security mechanisms. The paper deals with the definition, the reference model that describes the layers of the system, the areas of application and the trends of the Internet of Things. The requirements related to the security and privacy of systems and users as well as the most common security flaws in such systems are defined. The analysis and comparisons of security mechanisms of certain devices and protocols of the Internet of Things protocol stack are performed. The implementation of the system for monitoring the change of body temperature, its architecture and parts of the system are described, and a safety analysis of the entire system is performed.

Keywords: Internet of Things, Information security, ESP32, Wi-Fi, Bluetooth Low Energy, LoRaWAN, IP, TCP, UDP, HTTP, TLS, CoAP, MQTT, Analysis, Comparison